

УДК 007.52; 681.5

КП

№ держреєстрації 0109U001646

Інв. №

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ
(ХНУРЕ)

61166, м. Харків, пр. Леніна, 14
тел. (057) 70-21-425

ЗАТВЕРДЖУЮ
Проректор з наукової роботи
ХНУРЕ,
проф., д-р фіз.-мат. наук

_____ М.І.Сліпченко
«__» _____ 2011 р.

ЗВІТ
ПРО НАУКОВО-ДОСЛІДНУ РОБОТУ № 232

**«ТЕОРІЯ Й ПРОЕКТУВАННЯ ЕНЕРГОЗБЕРІГАЮЧИХ ЦИФРОВИХ
ОБЧИСЛЮВАЛЬНИХ СИСТЕМ НА КРИСТАЛАХ,
ЩО МОДЕЛЮЮТЬ І ПІДСИЛЮЮТЬ ФУНКЦІОНАЛЬНІ
МОЖЛИВОСТІ ЛЮДИНИ»**

(заключний)

Керівник НДР
Ректор ХНУРЕ
д.т.н., проф.

М.Ф. Бондаренко

2011

Рукопис закінчений «__» _____ 2011р.

Результати роботи розглянуті науково-методичною радою ХНУРЕ,
 Протокол № __ від «__» _____ 2011 р.

СПИСОК АВТОРІВ

Керівник НДР, ректор ХНУРЕ, д.т.н., проф.	М.Ф. Бондаренко
Відповідальний виконавець НДР, керівник розділу 232-1, декан факультету КІУ, д.т.н., проф.,	В.І. Хаханов (розділ 1)
Відповідальний виконавець розділу 232-1, проф. каф. АПОТ, д.т.н., проф.	С.В. Чумаченко (розділ 1)
Відповідальний виконавець розділу 232-2, пров. наук. співроб. каф. ПЗ ЕОМ, д.т.н., проф.	С.Ю. Шабанов-Кушнарєнко (розділ 3)
Керівник розділу 232-3 д.т.н., проф., зав. каф. ЕОМ	О.Г. Руденко (розділ 2)
Віповідальний виконавець розділу 232-3, пров. наук. співроб. каф. ЕОМ	О.М. Сотников (розділ 2)
Виконавець пров. наук. співроб. каф. ПЗ ЕОМ, д.т.н., проф.	Ю.П. Шабанов-Кушнарєнко (розділ 3)
Доц. каф. ПЗ ЕОМ, к.т.н.,	І.Д. Вєчірська (розділ 3)
Н.с. каф. АПОТ, к.т.н.	Д.Є. Шуклін (розділ 1)
С.н.с. каф. АПОТ к.т.н.	В.Б. Таранов (розділ 1)
Доц. каф. ЕОМ, к.т.н.	О.О. Торба (розділ 2)
С.н.с. каф. ЕОМ, к.т.н.	С.Г. Єлаков (розділ 2)
С.н.с. каф. ЕОМ, к.т.н.	А.О. Єлісеєв (розділ 2)
Проф. каф. ЕОМ, д.т.н.	С.Г. Удовєнко (розділ 2)
Проф. каф. АПОТ, д.т.н.	Є.І. Литвінова (розділ 1)
Пошукач каф. АПОТ	І.В. Філіпенко (розділ 1)
Зав. каф. АПОТ, д.т.н., проф.	Г.Ф. Кривуля (розділ 1)
Доц. каф. АПОТ, к.т.н.	О.С. Шкиль (розділ 1)
Н.с. каф. ПЗ ЕОМ, к.т.н.	А.В. Работягов (розділ 3)
М.н.с. каф. ПЗ ЕОМ	М.М. Вікарчук (розділ 3)

РЕФЕРАТ

Звіт 295 с., рис. 67, табл. 9, публікації: монографії 7, навчальні посібники та підручники 4, статті 114, тези 222, патенти 3.

За завданням НДР проведено аналіз стану проблеми, висвітлені основні задачі та проблемні питання, означені шляхи їх реалізації, наведено характеристику основних наукових та практичних результатів роботи. Наукові-практичним результатом, на одержання якого була спрямована НДР, є розробка теорії й проектування енергозберігаючих цифрових обчислювальних систем на кристалах, що моделюють і підсилюють функціональні можливості людини, шляхом імплементації в кристал паралельних синтезаторів та аналізаторів мови, швидкодійних модулів стиску та відтворення інформації, візуалізації та аналізу відображення.

Ключові слова.

Мультипроцесор, векторно-логічний аналіз та простір, критерій якості, діагностування несправностей пам'яті, процес-модель, алгебра предикатів, реляційні мережі мозкоподібних ЕОМ.

ЗМІСТ

Список умовних скорочень.....	6
Вступ.....	9
Розділ 1. Структура логічного асоціативного мозкоподібного енергозберігаючого мультипроцесора	10
1.1. Актуальність та спрямованість дослідження	10
1.2. Метрика кіберпростору для оцінювання розв'язку.....	12
1.3. Структури даних для пошуку, розпізнавання й прийняття рішення.....	23
1.4. Архітектура логічного асоціативного мультипроцесора.....	27
1.5. Інфраструктура векторно-логічного аналізу.....	32
1.6. Висновки до розділу 1.....	38
Розділ 2. Методи та архітектурні принципи швидкодіючих апаратно-програмних засобів попередньої обробки і класифікації зображень.....	41
2.1. Розробка архітектурних принципів систем обробки інформації в реальному часі на базі нейронної мережі СМАС.....	41
2.2. Розробка структур обчислювачів на базі генераторів сплайн функцій...113	
2.3. Розробка архітектури обчислювачів для кольірних перетворень зображень	164
2.4. Швидка попередня обробка та класифікація зображень дрібноструктурних об'єктів	196
2.5. Структури засобів швидкої обробки великих масивів інформації	213
2.6. Висновки до розділу 2.....	234
Розділ 3. Розробка теорії та принципів побудови мозкоподібних ЕОМ з реляційними мережами.....	236
3.1. Загальна характеристика дослідження.....	236
3.2. Опис процесу наукового дослідження.....	236
3.3. Отримані найважливіші наукові та науково-технічні результати.....	238
3.4. Відмінні риси і перевага отриманих результатів (продукції) над вітчизняними або зарубіжними аналогами чи прототипами.....	240

3.5. Висновки до розділу 3.....	242
Розділ 4. Інші результати, отримані та реалізовані в рамках НДР.....	244
4.1. Монографії та навчальні посібники.....	244
4.2. Патенти.....	245
4.3. Статті.....	245
4.4. Тези доповідей на конференціях.....	258
4.5. Договори з зацікавленими організаціями.....	280
4.6. Організація та проведення конференцій.....	281
4.7. Отримані нагороди.....	282
4.8. Отримані гранти.....	283
4.9. Дисертації та впровадження результатів.....	285
4.10. Виставки.....	289
4.11. Відкриття лабораторій.....	295

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

LAMP	– Logical Associative MultiProcessor
ЛАМП	– Логічний асоціативний мультипроцесор
slc	– Shift Left Bit Crowding
LUT	– Look Up Table
PLD	– Programmable Logic Device
CPU	– Central Processor Unit
НВІС	– Надвелика інтегральна схема
ППУ	– Пристрій послідовного управління
LP	– логічний процесор
UUT	– Unit Under Test
MUT	– Model Under Test
AD	– Address Decoder
SM	– Spare Memory
СМАС	– Cerebellar Model Articulation Controlle’
Сист	– Система
Інт	– Інтелект
Техн.	– Технологія
Онт.ПдО	– Онтологія ПдО
Зн.ПдО	– Знання про ПдО
ІС	– Інформаційні системи
ІТ	– Інформаційні технології
Онт.ІС	– Онтологія ІС
Відн	– Відносини
М	– Модель
Ф	– Функція
Х	– Характеристика
Ап.Пл	– Апаратна платформа
ППО	– Початковий перелік обмежень

БП	– Банк пам'яті
ВП	– Ведучий процесор
ПЯ	– Процесорне ядро
V-IRAM	– Інтелектуальна векторна архітектура
БПП	– Бібліотека прикладних програм
КМЯ	– Компілятор мови високого рівня
БОУП	– Блок обробки пакета
МКС	– Макросервер
ПГП	– Підсистема геометричних перетворень
ПСД	– Підсистема формування символічних даних
ПГД	– Підсистема формування графічних даних
ВУС	– Пристрій синхронізації
ВКП	– Відеоконтрольний пристрій
БС	– Блок сполучення з відеоконтрольним пристроєм
ОЗП	– Оперативний запам'ятовуючий пристрій
ЕОМ	– Електронна обчислювальна машина
ГА	– Генератор адреси
ПЕОМ	– Персональна електронна обчислювальна машина
ОЗПС	– Оперативний запам'ятовуючий пристрій кодів символів
ПУС	– Пристрій управління оперативним запам'ятовуючим пристроєм кодів символів
ФДР	– Формувач динамічних режимів виділення інформації
ОЗПГ	– Графічний оперативний запам'ятовуючий пристрій
БЗТ	– Блок збільшення точності відображення графічних інформаційних моделей
ПУГ	– Пристрій управління графічним оперативним запам'ятовуючим пристроєм
ПЗП	– Постійний запам'ятовуючий пристрій
АЦП	– Аналого-цифровий перетворювач
ЦАП	– Цифро-аналоговий перетворювач

ПЦОВ	– Пристрій цифрової обробки відеосигналу
СВІ	– Система відображення інформації
СФР	– Система функціональної регенерації
АП	– Адресний процесор
ЗП	– Запам'ятовуючий пристрій
БР	– Буфер регенерації
ФГА	– Функціональний генератор адрес
ПК	– Персональний комп'ютер
РЗ	– Регістр зрушення
БУ	– Блок управління
ГС	– Генератор символів
ПВІ	– Пристрій відображення інформації
ССВ	– Синтезуюча система візуалізації

ВСТУП

Метою проекту є розробка теорії й проектування енергозберігаючих цифрових обчислювальних систем на кристалах, що моделюють і підсилюють функціональні можливості людини, шляхом імплементації в кристал паралельних синтезаторів та аналізаторів мови, швидкодіючих модулів стиску та відтворення інформації, візуалізації та аналізу відображення.

Об'єктом дослідження є паралельна обробка інформації у природі і технічних пристроях.

Предмет дослідження – реляційна обробка інформації в природних об'єктах та її технічна реалізація штучного інтелекту в системи на кристалах.

Основна ідея проекту полягає в розробці та впровадженні методів, що ґрунтуються на алгебро-логічному інструментарії алгебри предикатів як основного засобу реалізації відношень довільної природи.

Завдання проекту: 1) Розробка інструментарію алгебри предикатів. 2) Розробка методів опису інформаційних об'єктів засобами алгебри предикатів. 3) Розробка методів побудови реляційних мереж мозкоподібних ЕОМ, зокрема, загального методу побудови реляційних мереж для довільних відношень. 4) Розробка нейро-фаззі компонентів для розпізнавання образів й прийняття інтелектуальних рішень. 5) Розробка вейвлет-трансформерів (JPEG) інформаційних потоків для бездротової передачі на відстані й зв'язку із зовнішнім середовищем. 6) Розробка мережних логічних мозкоподібних компонентів для керування, аналізу й синтезу інтелектуальних і образних функцій людини. 7) Створення образного робота, що сполучить всі сучасні технології (безпроводність, самовідновлюваність, наявність органів почуттів, Internet-керуваність). 8) Розробка образно-пошукового компоненту для виконання функцій точної навігації в Internet. 9) Захист програмно-апаратних компонентів від несанкціонованих доступів і руйнування.

РОЗДІЛ 1. СТРУКТУРА ЛОГІЧНОГО АСОЦІАТИВНОГО МОЗКОПОДІБНОГО ЕНЕРГОЗБЕРІГАЮЧОГО МУЛЬТИПРОЦЕСОРА

Пропонується інфраструктура забезпечення паралельного аналізу табличних або матричних форм асоціативних відношень для пошуку, розпізнавання й прийняття рішень в n -мірному векторному логічному просторі на основі використання запропонованої архітектури логічного асоціативного мозкоподібного енергозберігаючого мультипроцесора, орієнтованого на моделювання і підсилення функціональних можливостей людини. Розглядаються векторно-логічні процес-моделі актуальних прикладних завдань, у тому числі – вбудоване діагностування й відновлення працездатності компонентів цифрових систем на кристалах, де якість розв'язку оцінюється уведеною неарифметичною метрикою взаємодії булевих векторів.

1.1. Актуальність та спрямованість дослідження

Ідея дослідження – виключити з комп'ютера «великовагову» арифметику й трансформувати ресурси, що звільнилися, для створення інфраструктури векторно-логічних обчислень для швидкого пошуку, розпізнавання й прийняття рішень при аналізі інформаційного простору за допомогою примітивних операцій: and, or, not, xor. Спеціалізація комп'ютерного виробу, орієнтована на використання тільки логічних операцій, дає можливість істотно ($\times 100$) підвищити швидкодію розв'язку неарифметичних завдань. Виключення арифметичних операцій, використання паралелізму алгебри векторної логіки, мультипроцесорність архітектури створюють ефективну інфраструктуру, що поєднує математичну й технологічну культуру для розв'язування прикладних завдань. Ринкова привабливість логічного асоціативного мультипроцесора (Logical Associative MultiProcessor – LAMP) визначається тисячами старих і нових логічних за своєю природою задач, які в цей час вирішуються неефективно на надлишкових універсальних комп'ютерах з потужним арифметичним процесором. От деякі актуальні для ІТ-ринку проблеми: 1.

Аналіз, синтез і корекція синтаксичних і семантичних мовних конструкцій (реферування, виправлення помилок, оцінювання якості текстів). 2. Розпізнавання відео- і аудіо-образів на основі їхнього подання векторними моделями істотних параметрів у дискретному просторі. 3. Сервісне обслуговування складних технічних виробів і відновлення працездатності в процесі їхнього функціонування. 4. Тестування знань і експертне обслуговування об'єктів або суб'єктів для визначення їх валідності. 5. Ідентифікація об'єкта або процесу для прийняття рішення в умовах невизначеності. 6. Точний пошук заданої вектором параметрів інформації в Internet. 7. Цілевказівка у винищувачі або в автоматичній системі посадки лайнера, що працює у реальному мікросекундному діапазоні часу. 8. Розведення об'єктів за часом й за простором для диспетчерської служби аеропорту або оптимізація інфраструктури міського транспорту з метою виключення колізій. Практично всі перелічені завдання вирішуються в реальному масштабі часу, є ізоморфними за логічною структурою процес-моделей, що використовують сукупність взаємозалежних асоціативних таблиць. Для їхнього розв'язання необхідна спеціалізована апаратна платформа (LAMP), орієнтована на паралельне виконання процедур пошуку, розпізнавання й прийняття рішень, оцінюваних шляхом використання неарифметичного критерію якості.

Мета дослідження – істотне підвищення швидкодії процедур пошуку, розпізнавання й прийняття рішень шляхом мультипроцесорної й паралельної реалізації асоціативно-логічних векторних операцій для аналізу графових і табличних структур даних у векторному логічному просторі без використання арифметичних операцій.

Для досягнення поставленої мети необхідно розробити:

- 1) Неарифметичну метрику оцінювання векторно-логічних рішень у кібернетичному просторі.
- 2) Структури даних і процес-моделі розв'язання актуальних завдань.

3) Архітектуру логічного асоціативного мозкоподібного енергозберігаючого мультипроцесора й показати шляхи його практичного використання.

Об'єктом дослідження є інфраструктура процесів пошуку, розпізнавання й прийняття рішень у векторно-логічному просторі на основі використання алгебри векторної логіки, обчислювальної архітектури аналізу асоціативно-логічних структур даних і неарифметичного інтегрального критерію якості.

У процесі досліджень використані джерела науково-технічної інформації: асоціативно-логічні структури даних для розв'язання інформаційних завдань; апаратна платформа векторно-логічного аналізу інформації; моделі й методи дискретного аналізу й синтезу об'єктів кіберпростору.

1.2. Метрика кіберпростору для оцінювання розв'язку

Дискретний векторно-логічний простір – кіберпростір – сукупність взаємодіючих за відповідною метрикою інформаційних процесів і явищ, описуваних векторами (кортежами) логічні змінні й комп'ютерні системи, що використовують як носій, і мережі.

Метрика – спосіб виміру відстані в просторі між компонентами процесів або явищ, описаних векторами логічних змінних. Відстань у кіберпросторі це – хог-відношення між парою векторів, що позначають компоненти процесу або явища, що відрізняє його від кодової відстані за Хемінгом. Відстань, похідна (булева), ступінь зміни, розходження або близькості є ізоморфні поняття, пов'язані з визначенням відношення двох компонентів процесу або явища. Поняття близькості (відстані) компонентів у кіберпросторі є міра їхнього розходження. Процедури порівняння, виміру, оцінювання, розпізнавання, тестування, діагностування, ідентифікування, є спосіб визначення відношення при наявності не менш ніж двох об'єктів.

Компонент простору представлений k -мірним (двійковим) вектором $a = (a_1, a_2, \dots, a_j, \dots, a_k)$, $a_j \in \{0, 1\}$, де кожна його координата визначена у двійковому

алфавіті, 0 – «неправда», 1 – «істина». Нуль-вектор є k -мірний кортеж, всі координати якого дорівнюють нулю: $a_j = 0, j = \overline{1, k}$.

Метрика β кібернетичного простору визначається єдиною рівністю, що формує нуль-вектор для хог-суми відстаней d_i між ненульовим і кінцевим числом точок (об'єктів), замкнених у цикл:

$$\beta = \bigoplus_{i=1}^n d_i = 0, \quad (1.1)$$

де n – кількість (ціле число) відстаней між компонентами (векторами) простору, що становлять цикл $D = (d_1, d_2, \dots, d_i, \dots, d_n)$, d_i – є вектор відстані, що відповідає ребру циклу, який з'єднує два компоненти (вектори) a, b простору, що далі позначається без індексу як $d(a, b)$. Відстань між двома об'єктами (векторами) a і b є похідний вектор: $d(a, b) = (a_j \oplus b_j)_1^k$. Векторному значенню відстані відповідає норма – скалярна відстань за Хемінгом між двома векторами як число одиниць вектора $d(a, b)$. Інакше: метрика β векторного логічного двійкового простору є рівна нуль-вектору хог-сума відстаней між кінцевим числом крапок (вершин) графа, що утворюють цикл. Сума n -мірних двійкових векторів, що задають координати точок циклічної фігури, дорівнює нуль-вектору. Згортка простору в нуль-вектор становить інтерес для багатьох практичних завдань, включаючи: діагностування й виправлення помилок при передачі інформації з каналів зв'язку; пошук дефектів у цифрових виробках на основі двозначних і багатозначних таблиць несправностей. Крім того, на основі уведеної метрики можна дати більш формальне визначення кіберпростору, що є векторно-логічним, нормованим β -метрикою, де хог-сума відстаней між кінцевим числом точок циклу дорівнює нуль-вектору. Визначення метрики ставить в основу не елементи множини, але відношення, що дозволяє скоротити систему аксіом (тотожності, симетрії й транзитивного трикутного замикання) із

трьох до однієї й поширити її дію на як завгодно складні конструкції n -мірного логічного простору. Класичне завдання метрики для визначення взаємодії однієї, двох і трьох точок у векторному логічному просторі, є окремим випадком β -метрики при $i = 1, 2, 3$ відповідно:

$$M = \begin{cases} d_1 = 0 \leftrightarrow a = b; \\ d_1 \oplus d_2 = 0 \leftrightarrow d(a, b) = d(b, a); \\ d_1 \oplus d_2 \oplus d_3 = 0 \leftrightarrow d(a, b) \oplus d(b, c) = d(a, c). \end{cases} \quad (1.2)$$

Векторно-логічний транзитивний трикутник (1.2) має повну аналогію чисельному виміру відстані в метричному M -просторі, що задається системою аксіом, яка визначає взаємодію однієї, двох і трьох точок у будь-якому просторі:

$$M = \begin{cases} d(a, b) = 0 \leftrightarrow a = b; \\ d(a, b) = d(b, a); \\ d(a, b) + d(b, c) \geq d(a, c). \end{cases} \quad (1.3)$$

Специфіка аксіоми трикутника (1.3) метричного простору полягає в чисельному (скалярному) порівнянні відстаней трьох об'єктів, коли інтервальна невизначеність відповіді – дві сторони трикутника можуть бути більше або рівні третьої – малопридатна для визначення точної довжини останньої сторони. Векторно-логічний простір усуває даний недолік, повністю виключає ступінь невизначеності у бінарному відношенні детермінованих станів процесів або явищ. У цьому випадку чисельна невизначеність третьої сторони трикутника у векторному логічному просторі здобуває форму точного двійкового вектора, що характеризує відстань між двома об'єктами й обчислюється на основі знання відстаней двох інших сторін трикутника:

$$d(a, b) \oplus d(b, c) = d(a, c) \rightarrow d(a, b) \oplus d(b, c) \oplus d(a, c) = 0.$$

Приклад. Є п'ять точок у векторному просторі: (000111, 111000, 101010, 010101, 110011). Замикання цих точок у цикл дає наступні сторони-відстані в п'ятикутнику: (111111, 010010, 111111, 100110, 110100). Покоординатне додавання всіх векторів дає результат: (000000). Практична значимість даного факту полягає в можливості відновлення будь-якої відстані в замкнутому циклі, якщо відомі (n-1) сторона фігури. Для трикутника це означає відновлення третьої сторони за відомими двома. Якщо ж створити із трикутників замкнутий логічний простір, то можна заощадити 66% від обсягу даних, що формує всі відстані в логічному просторі.

Метрика β кібернетичного багатозначного векторно-логічного простору, є вектор, що дорівнює значенню \emptyset за всіма координатами, отриманий шляхом застосування симетричної різниці відстаней між кінцевим числом точок, які утворюють цикл:

$$\beta = \Delta_{i=1}^n d_i = \emptyset, \quad (1.4)$$

де кожна координата вектора, що відповідає об'єкту, визначена в алфавіті, який становить булеан на універсумі примітивів потужністю p : $a_j = \{\alpha_1, \alpha_2, \dots, \alpha_r, \dots, \alpha_m\}$, $m = 2^p$.

Рівність порожньому вектору симетричної різниці покоординатної теоретико-множинної взаємодії (1.4) підкреслює рівнозначність компонентів (відстаней), що беруть участь у формуванні рівняння, де єдина координатна операція $d_{i,j} \Delta d_{i+1,j}$, використовувана, наприклад, у чотиризначній моделі Кантора $A = \{0, 1, x, \emptyset\}$, $x = \{0, 1\}$, визначається відповідною таблицею:

Δ	0	1	x	\emptyset	\cap	0	1	x	\emptyset	\cup	0	1	x	\emptyset	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td>a</td> <td>0</td> <td>1</td> <td>x</td> <td>\emptyset</td> </tr> <tr> <td>\tilde{a}</td> <td>1</td> <td>0</td> <td>\emptyset</td> <td>x</td> </tr> </table>	a	0	1	x	\emptyset	\tilde{a}	1	0	\emptyset	x
a	0	1	x	\emptyset																					
\tilde{a}	1	0	\emptyset	x																					
0	\emptyset	x	1	0	0	0	\emptyset	0	\emptyset	0	0	x	x	0											
1	x	\emptyset	0	1	1	\emptyset	1	\emptyset	1	1	x	1	x	1											
x	1	0	\emptyset	x	x	0	1	x	\emptyset	x	x	x	x	x											
\emptyset	0	1	x	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	0	1	x	\emptyset											

Тут також наведені таблиці істинності для інших базових теоретико-множинних операцій (перетинання, об'єднання, доповнення), далі використовуваних за текстом. Число примітивних символів, що утворюють замкнутий щодо теоретико-множинних координатних операцій алфавіт, може бути збільшено. При цьому потужність алфавіту (булеану) визначається виразом $m = 2^p$, де p – число примітивів. Для практичного використання уведеної метрики кіберпростору далі пропонується доказовий перехід від чисельної характеристики бінарного відношення об'єктів, що поєднує три скалярні оцінки їхньої взаємодії до чисто векторно-логічного критерію якості відношення двох об'єктів.

Основні припущення й визначення. Вхідний вектор $m = (m_1, m_2, \dots, m_j, \dots, m_k)$, $m_j \in \{0, 1, x\}$ і аналізований об'єкт $A = (A_1, A_2, \dots, A_j, \dots, A_k)$, $A_j \in \{0, 1, x\}$, що також представлений вектором, мають однакову розмірність k . Ступінь приналежності m -вектора до A позначається як $\mu(m \in A)$. Існує 5 типів координатної теоретико-множинної Δ -взаємодії двох векторів $m \Delta A$ (рис. 1.1). Вони формують всі примітивні варіанти реакції узагальненої системи пошуку, розпізнавання й прийняття рішення на вхідний вектор-запит.

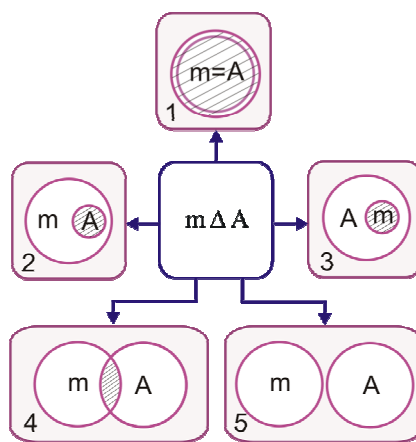


Рисунок 1.1 – Результати взаємодії двох векторів

У технологічній галузі знань – технічній діагностиці – згадана послідовність дій ізоморфна маршруту: пошук дефектів, їхнє розпізнавання, прийняття рішення на відновлення працездатності. Дані стадії технологічного маршруту мають потребу в метриці оцінювання рішень для вибору оптимального варіанта.

Визначення. Інтегральна теоретико-множинна метрика для оцінювання якості запиту є функція взаємодії багатозначних за координатами векторів $m \Delta A$, що визначається середньою сумою трьох параметрів: кодова відстань $d(m, A)$, функція приналежності $\mu(m \in A)$ й функція приналежності $\mu(A \in m)$:

$$Q = \frac{1}{3} [d(m, A) + \mu(m \in A) + \mu(A \in m)],$$

$$d(m, A) = \frac{1}{n} [n - \text{card}\{(i : m_i \cap A_i = \emptyset, i = 1, \dots, k)\}]; \mu(m \in A) = 2^{c-a}; \mu(A \in m) = 2^{c-b};$$

$$a = \text{card}\{(i : A_i = x, i = 1, \dots, k)\}; b = \text{card}\{(i : m_i = x, i = 1, \dots, k)\}; c = \text{card}\{(i : m_i \cap A_i = x, i = 1, \dots, k)\};$$
(1.6)

Пояснення. Перетинання (об'єднання) векторів – є векторна операція, заснована на відповідних координатних теоретико-множинних операціях. Операції координатного перетинання й об'єднання (1.6) визначені в алфавіті Кантора $A = \{0, 1, x = \{0, 1\}, \emptyset\}$. Нормування параметрів дозволяє оцінити рівень взаємодії векторів у чисельному інтервалі $[0, 1]$. Якщо зафіксовано граничне максимальне значення кожного параметра, рівне 1, то вектори рівні між собою. Мінімальна оцінка, $Q = 0$, фіксується у випадку повної розбіжності векторів за всіма n координатами. Якщо $m \cap A = m$ й потужність покоординатного перетинання дорівнює половині потужності простору вектора A , то функції приналежності і якості дорівнюють:

$$\mu(m \in A) = \frac{1}{2}; \mu(A \in m) = 1; d(m, A) = 1; Q(m, A) = \frac{5}{2 \times 3} = \frac{5}{6}.$$

Аналогічне значення буде мати параметр Q , якщо $m \cap A = A$ й потужність покоординатного перетинання дорівнює половині потужності простору вектора m . Тут простір вектора є функція від числа координат ω , рівних x : $q = 2^\omega$. Якщо $\text{card}(m \cap A)$ – потужність покоординатного перетинання дорівнює половині потужностей просторів векторів A і m , то функції приналежності мають значення:

$$\mu(m \in A) = \frac{1}{2}; \mu(A \in m) = \frac{1}{2}; d(m, A) = 1; Q(m, A) = \frac{1}{3} \times \left(\frac{1}{2} + \frac{1}{2} + 1 \right) = \frac{2}{3}.$$

Наприклад, дано два вектори: $A = (XXX10)$ і $m = (XX0X0)$. Їхнє перетинання дорівнює $(XX010) = \{00010, 01010, 10010, 11010\}$. Інакше, потужність результуючого простору дорівнює чотирьом двійковим векторам або половині потужностей вихідних двійкових векторів.

Варто помітити, якщо перетинання двох векторів дорівнює порожній множині $\exists i(m_i \cap A_i) = \emptyset$, то кількість загальних точок (двійкових векторів) при перетинанні двох просторів, формованих двома векторами, дорівнює нулю.

Ціль введення векторно-логічного критерію якості рішення полягає в істотному підвищенні швидкодії при підрахунку оцінки Q взаємодії компонентів (векторів) m і A при аналізі асоціативних структур даних шляхом використання тільки векторних операцій, виконуваних одночасно (паралельно) над всіма розрядами. З урахуванням ізоморфізму теоретико-множинних і логічних операцій арифметичний критерій (1.6) без усереднення функцій приналежності й кодової відстані можна трансформувати до виду:

$$\begin{aligned} Q &= d(m, A) + \mu(m \in A) + \mu(A \in m), \\ d(m, A) &= \text{card}(\{i : m_i \oplus A_i = U, i = 1, \dots, k\}); \\ \mu(m \in A) &= \text{card}(\{i : A_i = U, i = 1, \dots, k\}) - \text{card}(\{i : m_i \oplus A_i = U, i = 1, \dots, k\}); \\ \mu(A \in m) &= \text{card}(\{i : m_i = U, i = 1, \dots, k\}) - \text{card}(\{i : m_i \oplus A_i = U, i = 1, \dots, k\}); \\ U &= \begin{cases} 1 \leftarrow \{m_i, A_i\} \in \{0, 1\}; \\ x \leftarrow \{m_i, A_i\} \in \{0, 1, x\}. \end{cases} \end{aligned} \quad (1.7)$$

Якщо вектори m і A – двійкові за всіма координатами, то змінна $U=1$ і обчислення проводяться за правилами двійкової \oplus -операції. Якщо вектори m і A визначені у трійковому алфавіті, то змінна $U=x$ ініціює обчислення на основі використання теоретико-множинної операції симетричної різниці Δ (1.7). Введення змінної U дає можливість піти від написання двох формул критерію залежно від значності алфавіту опису координат взаємодіючих векторів. Представлені в (1.7) векторні логічні операції $(\wedge, \vee, \oplus, \neg)$ ізоморфні теоретико-множинним $(\cap, \cup, \Delta, \sim)$. При цьому теоретико-множинні координатні операції, що відповідають даним логічним, визначені раніше на багатозначному алфавіті Кантора у виразі (1.5). Перший компонент (1.7), що становить критерій, формує ступінь розбіжності k -мірних векторів – кодова відстань, шляхом виконання операції хог, другий і третій визначають ступені неприналежності результату кон'юнкції до числа одиниць кожного із двох взаємодіючих векторів. Поняття приналежності й неприналежності є взаємодоповнюючими, але в цьому випадку є більш технологічним обчислити неприналежність. Отже, необхідний критерій якості дорівнює нулю (за всіма координатами), коли два вектори рівні між собою. Оцінка якості взаємодії двох двійкових векторів погіршується в міру зростання критерію від нуля до одиниці. Для того, щоб остаточно виключити арифметичні операції при підрахунку векторно-логічного критерію якості, необхідно логічно об'єднати три оцінки (1.7) в одну:

$$\begin{aligned}
 Q &= d(m, A) \vee \mu(m \in A) \vee \mu(A \in m) = \\
 &= (m \oplus A) \vee (A \wedge \overline{m} \wedge \overline{A}) \vee (m \wedge \overline{m} \wedge \overline{A}) = \\
 &= (m \oplus A) \vee [A \wedge (\overline{m} \vee \overline{A})] \vee [m \wedge (\overline{m} \vee \overline{A})] = \\
 &= (m \oplus A) \vee [(A \wedge \overline{m}) \vee (A \wedge \overline{A}) \vee (m \wedge \overline{m}) \vee (m \wedge \overline{A})] = \\
 &= [(A \wedge \overline{m}) \vee (m \wedge \overline{A})] \vee [(A \wedge \overline{m}) \vee (A \wedge \overline{A}) \vee (m \wedge \overline{m}) \vee (m \wedge \overline{A})] = \\
 &= (A \wedge \overline{m}) \vee (m \wedge \overline{A}) \vee (A \wedge \overline{m}) \vee (A \wedge \overline{A}) \vee (m \wedge \overline{m}) \vee (m \wedge \overline{A}) = \\
 &= m \oplus A.
 \end{aligned}$$

Процедура обчислення векторного критерію якості залежить від значності алфавіту:

$$Q' = \begin{cases} m \oplus A \leftarrow \{m_i, A_i\} \in \{0,1\}; \\ m \Delta A \leftarrow \{m_i, A_i\} \in \{0,1,x\}. \end{cases} \quad (1.8)$$

Для двійкового алфавіту таблиця істинності координатної хог-операції має вигляд:

\oplus	0	1
0	0	1
1	1	0

У другому випадку, коли алфавіт опису координат має три значення, обчислення симетричної різниці здійснюється відповідно до Δ -операції, представленої в (1.5).

Критерій якості Q однозначно визначає три форми взаємодії двох будь-яких об'єктів в n -мірному векторному логічному просторі: відстань і дві функції приналежності. При ненульовій відстані за Хемінгом функції приналежності дорівнюють нулю, оскільки простори двох векторів у цьому випадку не перетинаються. У протилежному випадку – кодова відстань, рівна нулю – взаємодія об'єктів оцінюється за функціями приналежності. Збільшення числа нулів підвищує критерій якості, а збільшення кількості одиниць обумовлює погіршення якості взаємодії за відповідними булевими змінними. Критерій якості $Q = m \oplus A$ погоджується з уведеною вище метрикою оцінювання відстані або взаємодії об'єктів у векторно-логічному просторі, а також має тривіальну обчислювальну процедуру для оцінювання рішень, пов'язаних з аналізом і синтезом інформаційних об'єктів. Дійсно, векторний логічний простір не повинен мати метричної відстані й чисельних критеріїв якості, що включають арифметичні операції на скалярних величинах.

Для порівняння критеріїв якості необхідно визначити число одиниць у кожному векторі без виконання операцій підсумовування. Це можна зробити за

допомогою регістра зсуву (рис. 1.2), що дозволяє за один такт виконати процедуру slc (shift left bit crowding) - зсув вліво з одночасним ущільненням всіх одиничних координат n-розрядного двійкового вектора.

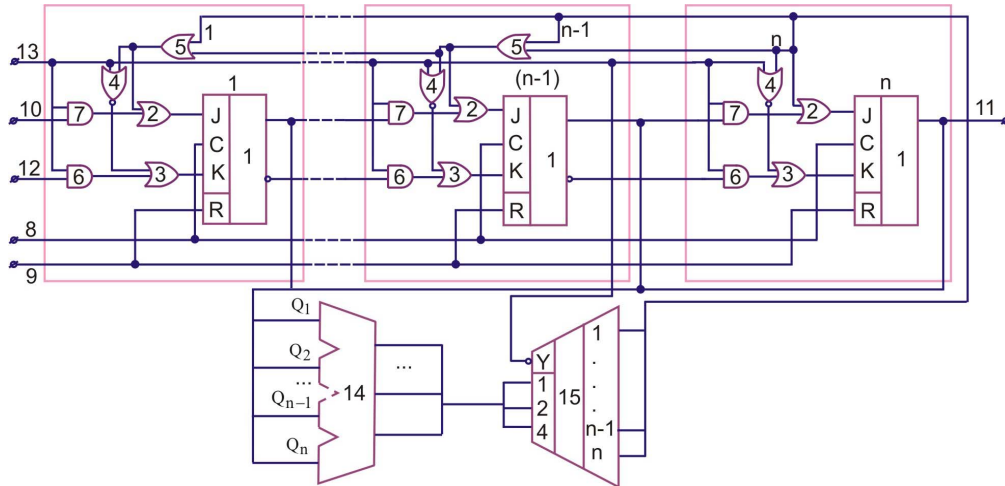


Рисунок 1.2 – Регістр зсуву й ущільнення одиниць

Після процедури стиску номер правого одиничного біта ущільненої серії одиниць формує значення критерію якості взаємодії векторів. Але в цьому випадку таке число є данина скалярній оцінці бінарного відношення, що потрібна лише людині як інформація для порівняння пропонованої інфраструктури щодо існуючих технологій. У практичних завданнях така оцінка втрачає свій зміст при виборі (квазі-) оптимальної альтернативи. Векторна оцінка є більш зручною для обчислювача, що визначає кращий розв'язок без участі в даному процесі користувача. Для наборів $m = (110011001100)$, $A = (000011110101)$ визначення якості їхньої взаємодії за формулами (1.8) представлено у вигляді наступних процедур (нульові координати відзначені точками):

m	1 1 . . 1 1 . . 1 1 . .
A 1 1 1 1 . 1 . 1
$Q^* = m \oplus A$	1 1 1 1 1 . . 1
$Q(m,A)$	1 1 1 1 1 1

Тут сформована оцінка взаємодії векторів i , що саме головне, одиничні координати рядка $Q^* = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$ ідентифікують всі істотні змінні, за якими взаємодія векторів не відповідає критерію якості. Процедура стиску для одержання $Q(m, A)$ не означає втрату інформативної векторної оцінки $Q^* = m(A)$. Результат стиску дозволяє лише зробити вибір кращого із двох або більше розв'язків шляхом порівняння сумарного числа одиниць, що формують скалярні оцінки критеріїв. Де їх менше, там розв'язок кращий.

Як вибрати кращий розв'язок, якщо їх декілька?

Процес-модель пошуку оцінки кращого розв'язку з мінімальним числом одиничних координат з більш, ніж двома альтернативами, представлена на рис.

1.3. Вона включає наступні операції:

1) Спочатку у вектор-результат Q , де буде збережений кращий розв'язок, заносяться одиничні значення в усі координати (гірший розв'язок) і одночасно здійснюється операція slc зрушення вліво з ущільненням одиниць поточного вектора Q_i .

2) Виконується порівняння двох векторів: Q і чергової оцінки Q_i зі списку розв'язків.

3) Реалізується векторна операція and ($Q \wedge Q_i$), а результат порівнюється з вектором Q , що дає можливість змінити його, якщо вектор Q_i має менше число одиничних значень.

4) Процедура пошуку оцінки кращого розв'язку повторюється n раз.

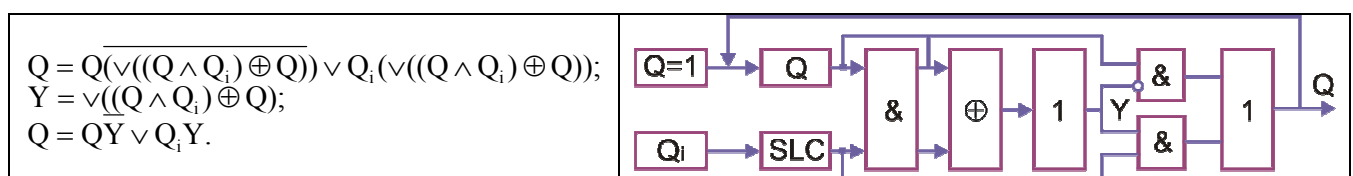


Рисунок 1.3 – Процес-модель вибору розв'язку

Тут or -оператор редукції (після хог елемента) формує двійковий однобітовий розв'язок на основі логічної операції or над n розрядами критерію

якості. Для двох двійкових векторів – критеріїв якості – процедура вибору кращого з них представлена нижче:

$Q_1(m, A) = (6,12)$	1 1 1 1 1 1
$Q_2(m, A) = (8,12)$	1 1 1 1 1 1 1 1
$Q_1(m, A) \wedge Q_2(m, A)$	1 1 1 1 1 1
$Q_1(m, A) \oplus Q_1(m, A) \wedge Q_2(m, A)$
$Q(m, A) = Q_1(m, A)$	1 1 1 1 1 1

Тут чисельні оцінки представлені як пояснювальна інформація для користувача, що більш технологічно сприймає числа й символи. Таким чином, векторний логічний критерій якості взаємодії об'єктів у кіберпросторі дозволяє одержувати оцінку пошуку, розпізнавання й прийняття рішення з високою швидкістю логічних паралельних операцій, що особливо істотно для критичних систем реального часу.

1.3. Структури даних для пошуку, розпізнавання й прийняття рішення

Метрика якості, представлена в (1.8), дає можливість оцінювати близькість просторових об'єктів друг до друга або їхню взаємодію. Практичним прикладом корисності векторно-логічного критерію якості може служити стрілянина за мішенню, що ілюструється вище наведеними діаграмами (див. рис. 1.1) взаємодії векторів: 1) Снаряд потрапив точно у мішень й уразив її повністю; 2) Мішень уражена необґрунтовано більшим калібром снаряда; 3) Калібр снаряда недостатній для поразки великої мішені; 4) Неефективний і неточний постріл снарядом великого калібру; 5) Снаряд пролетів повз мішені. Процес-модель взаємодії $P(m, A)$ супроводжується критерієм якості, що оцінює не тільки влучення або промах, але й ефективність використання калібру зброї. Аналітична форма запису узагальненої процес-моделі для вибору кращої взаємодії вхідного запиту m із системою логічних асоціативних відношень має вигляд:

$$\begin{aligned}
P(m, A) &= \min_{i=1}^n Q_i(m \Delta A_i) = \vee[(Q_i \wedge_{j=1, n}^{j \neq i} Q_j) \oplus Q_i] = 0; \\
Q(m, A) &= (Q_1, Q_2, \dots, Q_i, \dots, Q_n); A = (A_1, A_2, \dots, A_i, \dots, A_n); \\
\Delta &= \{\text{and, or, xor, not, slc, nop}\}; A_i = (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is}); \\
A_{ij} &= (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{ijmsq}); m = (m_1, m_2, \dots, m_r, \dots, m_q). \\
Q_i &= d(m, A_i) \vee \mu(m \in A_i) \vee \mu(A_i \in m); d(m, A_i) = m \oplus A_i; \\
\mu(m \in A_i) &= A_i \wedge \overline{m} \wedge A_i; \mu(A_i \in m) = m \wedge \overline{m} \wedge A_i.
\end{aligned} \tag{1.9}$$

Тут вираз $P(m, A)$, що визначає функціональність, можна представити як аналітичну модель обчислювального процесу у вигляді висловлення, що мінімізує критерій якості; структури даних представлені у вигляді сукупності таблиць $A = (A_1, A_2, \dots, A_i, \dots, A_m)$, які логічно взаємозалежать між собою; кожна таблиця задається впорядкованою сукупністю вектор-рядків асоціативної таблиці $A_i = (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is})$ явних розв'язків, а рядок $A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{ijmsq})$ являє собою істинне висловлення. Функціонал, представлений у вигляді таблиці, немає постійних у часі вхідних і вихідних змінних. Рівнозначність всіх змінних у векторі $A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{ijmsq})$ створює однакові умови їхнього існування. Це означає інваріантність розв'язку задач прямої і зворотної імплікації в просторі $A_i \in A$. Асоціативний вектор A_{ij} визначає собою явни розв'язок, де кожна змінна задається в кінцевому, багатозначному й дискретному алфавіті $A_{ijr} \in \{\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_k\} = \beta$. Взаємодія $P(m, A)$ вхідного вектора-запиту $m = (m_1, m_2, \dots, m_r, \dots, m_q)$ з множиною $A = (A_1, A_2, \dots, A_i, \dots, A_m)$ формує розв'язок з вибором кращого з них за мінімальним критерієм якості:

$$P(m, A) = \min Q_i[m \wedge (A_1 \vee A_2 \vee \dots \vee A_i \vee \dots \vee A_m)].$$

Конкретна взаємодія вершин (таблиць) графа між собою створює функціональність $A = (A_1, A_2, \dots, A_i, \dots, A_m)$, що може бути оформлена в наступні структури:

1) Єдина асоціативна таблиця, що містить всі розв'язки логічного завдання в явному вигляді. Перевага – максимальна швидкодія паралельного асоціативного пошуку розв'язку за таблицею. Недолік – максимально висока апаратна складність розміщення таблиці великої розмірності.

2) Деревоподібна (графова) структура бінарних відношень між функціональними примітивами, кожний з яких формує таблицю істинності для незначної кількості змінних. Перевага – максимально низька апаратна складність розв'язку завдання. Недолік – мінімальна швидкодія послідовного асоціативного пошуку розв'язку за деревом.

3) Компромісна графова структура логічно зрозумілих для користувача відношень між примітивами, кожний з яких формує таблицю істинності для логічно сильно взаємозалежних змінних. Перевага – висока швидкодія паралельного асоціативного пошуку розв'язків за мінімальним числом таблиць, що становлять граф, а також порівняно невисока апаратна складність розв'язку завдання. Недолік – зниження швидкодії через послідовну логічну обробку графової структури явних розв'язків, знайдених у таблицях, і необхідність введення додаткової змінної для логічного зв'язування таблиць (вершин) у граф відношень. Розбивка однієї таблиці (асоціативної пам'яті) на k частин приводить до зменшення апаратних витрат, виражених у компонентах (лутах) (LUT – Look Up Table) програмувальної логічної матриці. Кожна комірка пам'яті створюється за допомогою чотирьох лутів. З огляду на асоціативну матрицю, що можна представити квадратом зі стороною n , сумарні апаратні витрати $Z(k)$ пам'яті для зберігання даних і час $T(k)$ аналізу логічного асоціативного графа функціонально залежать від числа k розбивок таблиці або числа вершин графа:

$$Z(k) = k \times \frac{1}{4} \times \left(\frac{n}{k}\right)^2 + h = \frac{n^2}{4 \times k} + h, \quad (h = \{n, \text{const}\}); \quad (1.10)$$

$$T(k) = \frac{4 \times k}{t_{\text{clk}}} + \frac{4}{t_{\text{clk}}} = \frac{4}{t_{\text{clk}}}(k+1), \quad (t_{\text{clk}} = \text{const}).$$

Тут h – витрати на загальну схему керування системою асоціативних пам'ятей. Платою за зменшення апаратури є зниження швидкодії обробки структури пам'ятей або збільшення часу аналізу компонентів графа. Період обробки однієї асоціативної пам'яті представлений циклом з 4-х синхроімпульсів. Число розбивок k пропорційно збільшує кількість тактів у гіршому варіанті послідовного з'єднання пам'ятей. Доданок $\frac{4}{t_{\text{clk}}}$ в (1.10) задає час, необхідний для підготовки даних на вході обчислювальної структури, а також для їхнього декодування на виході. Функціональні залежності апаратних витрат і часу аналізу графа асоціативних пам'ятей від числа вершин або розбивок представлені на рис. 1.4.

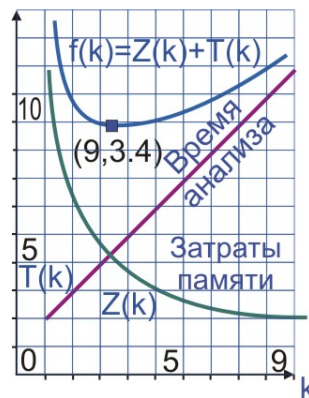


Рисунок 1.4 – Функції апаратури й часу від числа розбивок

Узагальнена функція ефективності графової структури від числа вершин

$$f[Z(k), T(k)] = Z(k) + T(k) = \left(\frac{n^2}{4 \times k} + h\right) + \left(\frac{4}{t_{\text{clk}}}(k+1)\right) \quad (1.11)$$

дозволяє визначити оптимальну розбивку сукупного й наперед заданого обсягу асоціативної пам'яті. У випадку, представленому на рис. 1.4, краща розбивка є мінімум адитивної функції, який визначається значенням k , що обертає похідну функції в нуль: $n \times n = 600 \times 600$, $h = 200$, $t_{\text{clk}} = 4$, $k = 4$. Запропонована структура асоціативних таблиць і уведений критерій якості одержуваних розв'язків є основою для розробки спеціалізованої мультипроцесорної архітектури, орієнтованої на паралельне виконання векторних логічних операцій.

1.4. Архітектура логічного асоціативного мультипроцесора

Для аналізу великих інформаційних обсягів логічних даних існує трохи практично орієнтованих технологій:

1. Використання робочої станції для послідовного програмування задачі, де вартість її розв'язання, а також часові витрати дуже високі.

2. Розробка спеціалізованого паралельного процесора на основі PLD. Високий паралелізм обробки інформації компенсує порівняно низьку в порівнянні з CPU тактову частоту. Такий схемотехнічний розв'язок з можливістю перепрограмування є за продуктивністю виграшним варіантом. Недолік – відсутність гнучкості, що властива програмним методам розв'язання логічних задач, і висока вартість реалізації системи на кристалі PLD при великих обсягах промислового випуску виробу.

3. Кращий розв'язок може бути отриманий в результаті об'єднання переваг центрального процесорного пристрою, програмувальної логічної матриці й замовленої НВІС, таких як гнучкість програмування, можливість коректування вихідних кодів; мінімальне число команд і прості схемотехнічні розв'язки апаратної реалізації мультипроцесора; розпаралелювання логічних процедур на структурі однобітових процесорів. Реалізація обчислювача в кристалі замовленої НВІС дає можливість одержувати максимальну тактову частоту, мінімальну вартість мікросхеми при великих обсягах випуску виробу, низьке енергоспоживання. Об'єднання переваг перерахованих технологій визначає базову конфігурацію обчислювача, що має сферичну структуру (рис.

1.5), що складається з 16-ти векторних секвенсорів – пристроїв послідовного управління (ППУ), кожний з яких, включаючи граничні елементи, з'єднаний зі вісьма сусідніми. Прототипом даного обчислювача є процесор PRUS.

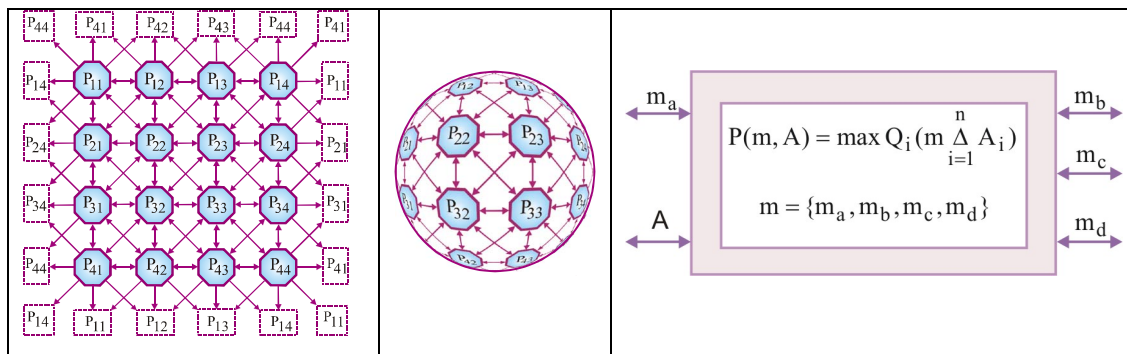


Рисунок 1.5 – Макроархітектура мультипроцесора та інтерфейс

Занесення інформації до процесора виконується за класичною схемою процесу проектування, за винятком стадії розміщення й трасування, що замінюється фазою розподілу програм і даних між всіма логічними біт-процесорами, що працюють паралельно. Компілятор забезпечує розподіл даних по процесорах, задає час формування розв'язку на виході кожного з них, а також планує передачу отриманих результатів іншому процесору.

Логічний асоціативний мультипроцесор (ЛАМП) – це ефективна мережа процесорів, що обробляє дані й забезпечує обмін інформацією між компонентами мережі в процесі їхнього розв'язання. Проста схемотехніка кожного процесора дозволяє ефективно обробляти надвеликі масиви, що нараховують мільйони біт інформації, затрачаючи на це в сотні разів менше часу в порівнянні з універсальним процесором.

Базовий осередок – векторний процесор для обчислювача – може бути синтезований на 200-х вентилях, що дає можливість мережу, яка містить 4096 обчислювачів, легко реалізувати в кристалі замовленої НВІС, використовуючи сучасну кремнієву технологію. Оскільки витрати пам'яті для зберігання даних досить незначні, обчислювач може бути використаний при проектуванні систем

керування в таких областях людської діяльності, як промислове виробництво, захист інформації, медицина, штучний інтелект, космонавтика, геологія, метеорологія. Особливий інтерес обчислювач представляє для цифрової обробки даних, розпізнавання образів і криптоаналіза. Однак основне призначення ЛАМП – одержання квазіоптимального розв’язку у задачі пошуку й (або) розпізнавання з використанням компонентів архітектури, орієнтованих на виконання векторних логічних операцій:

$$P(m, A) = \min_{i=1}^n Q_i(m \Delta A_i), m = \{m_a, m_b, m_c, m_d\}. \quad (1.12)$$

Інтерфейс системи, що відповідає даному функціоналу (1.12), представлений на рис. 1.5. Компоненти $\{A, m_a, m_b, m_c, m_d\}$ можуть бути як вхідними, так і вихідними. Двонаправлена деталізація інтерфейсу пов’язана з інваріантністю відношення всіх змінних, векторів, *A-матриці* й компонентів до входів і (або) виходів архітектури. Тому структурна модель системи обчислювача може бути використана для розв’язання будь-яких задач прямої і зворотної імплікації в дискретному логічному просторі, у чому й полягає її відмінність від концепції автоматної моделі обчислювального пристрою з вираженими входами й виходами. Компоненти або регістри $m = (m_a, m_b, m_c, m_d)$ використовуються для одержання розв’язку у вигляді буферних, вхідних і вихідних векторів, а також для ідентифікації оцінки якості задоволення вхідного запиту.

Один з можливих варіантів архітектури ЛАМП представлений на рис. 1.6. Основним компонентом є матриця $P = [P_{ij}], (i, j = \overline{1,4})$, що містить 16 вектор-процесорів, кожний з яких призначений для виконання п’яти логічних векторних операцій над пам’яттю даних, представленої у вигляді таблиці A , розмірністю $(m \times n)$.

У блоці інтерфейсу відбувається обмін даними й завантаження програми обробки даних у відповідну пам'ять команд. Блок керування ініціює виконання команд логічної обробки даних і синхронізує функціонування всіх компонентів мультипроцесора. Блок *IP* призначений для сервісного обслуговування всіх модулів, діагностування дефектів і відновлення працездатності компонентів і пристрою в цілому. Елементарний логічний асоціативний процесор або ППУ (див. рис. 1.5), що входить до складу обчислювача, містить логічний процесор *LP*, асоціативну (пам'ять) *A-матрицю* для паралельного виконання базових операцій, блок векторів *m*, призначений для паралельного обслуговування рядків і стовпців *A-матриці*, а також обміну даними в процесі обчислень, пам'ять прямого доступу *CM*, що зберігає команди програми обробки інформації, автомат керування *CU* виконанням логічних операцій, інтерфейс *I* зв'язку ППУ з іншими елементами й пристроями ЛАМП.

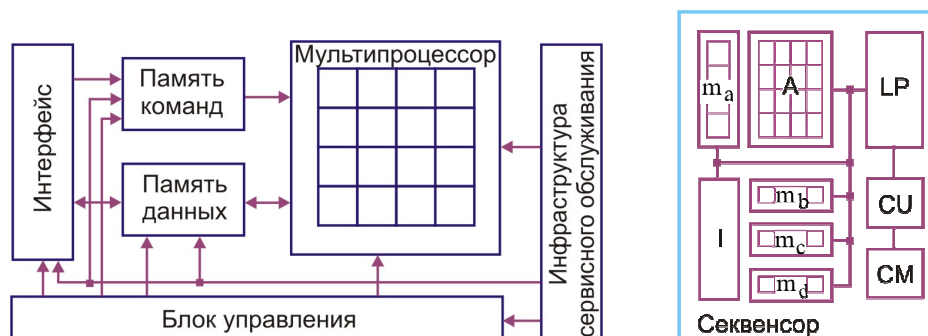


Рисунок 1.6 – Архітектура ЛАМП і структура ППУ

Логічний процесор (LP) (рис. 1.7) виконує п'ять операцій (and, or not, xor, slc), що є базовими для створення алгоритмів і процедур інформаційного пошуку й оцінювання розв'язку. Модуль LP має мультиплексор, комутуючий один з п'яти операндів з обраним логічним векторним оператором. Сформований результат через мультиплексор (елемент or) заноситься в один із чотирьох операндів, що обирається відповідною адресою.

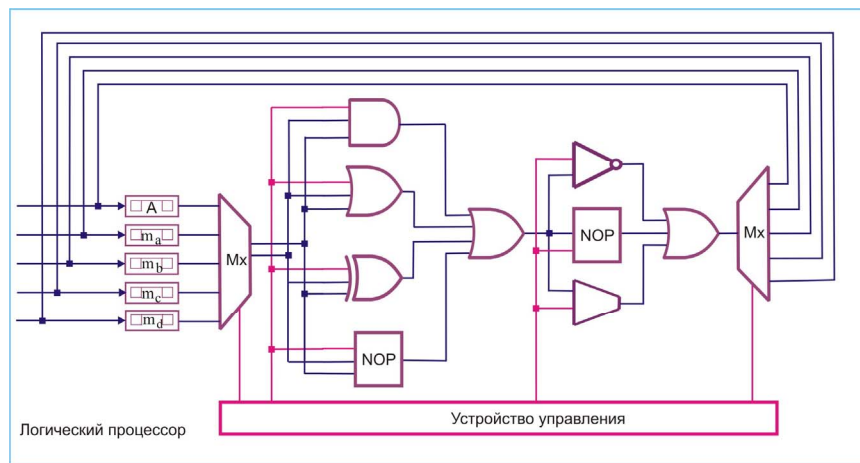


Рисунок 1.7 – Структура блоку логічних обчислень

Особливості реалізації логічного процесора полягають у наявності трьох бінарних (and, or, xor) і двох унарних (not, slc) операцій. Останні можна приєднувати до такту обробки регістрових даних, вибравши одну із трьох операцій (not, slc, nop – немає операції). Для підвищення ефективності роботи логічного пристрою вводяться два елементи з порожньою операцією. Якщо необхідно виконати тільки унарну операцію, то на рівні бінарних команд варто вибрати nop, що практично означає передачу даних через повторювач до другого рівня унарних операцій. Всі операції в LP – регістрові або регістрово-матрицеві. Останні призначені для аналізу вектор-рядків таблиці при використанні вхідного *m*-вектора як запиту для точного пошуку інформації. У блоці логічних обчислень припустимо наступне сполучення операцій і операндів:

$$C = \begin{cases} \{m_a, m_b, m_c, m_d\} \Delta A_i; \\ \{m_a, m_b, m_c, m_d\} \Delta \{m_a, m_b, m_c, m_d\}; \\ \{\text{not, nop, slc}\} \{m_a, m_b, m_c, m_d, A_i\}. \end{cases}$$

$$\Delta = \{\text{and, or, xor}\}.$$

Реалізація всіх векторних операцій блоку логічних обчислень, виконуваних з тактовою частотою 100 МГц, для одного ППУ в середовищі

Verilog з наступною післясинтезною реалізацією в кристалі програмувальної логіки Virtex 4, Xilinx містить 2400 еквівалентних вентилів.

1.5. Інфраструктура векторно-логічного аналізу

Інфраструктура – сукупність моделей, методів і засобів опису, аналізу й синтезу структур даних для рішення функціональних завдань. Модель (системна) - сукупність взаємозалежних, певних у просторі й часі компонентів, що із заданою адекватністю описує процес або явище й використовується для досягнення поставленої мети при наявності обмежень і метрики оцінювання якості розв’язку. Тут обмеження є апаратурні витрати, час розробки й виробництва до появи виробу на ринку (time-to-market), що повинні бути мінімізованими. Метрика оцінювання розв’язку при використанні моделі визначена двійковим логічним вектором у дискретному булевому просторі. Концептуальна модель обчислювального виробу представлена сукупністю керуючих і операційного автоматів. У моделі функціональності використана ієрархічна технологія створення цифрових систем з локальною синхронізацією окремих модулів і одночасно глобальною асинхронністю функціонування всього пристрою.

Для деталізації структури векторного процесора й ППУ далі розглянуто аналітичні й структурні процес-моделі, що виконують аналіз *A-матриці* за стовпцями або рядками. Перша з них представлена на рис. 1.8 і призначена для визначення множини припустимих розв’язків щодо вхідного запиту m_b , друга (рис. 1.9) здійснює пошук оптимального розв’язку на множині рядків, знайдених за допомогою першої моделі в результаті їхнього аналізу.

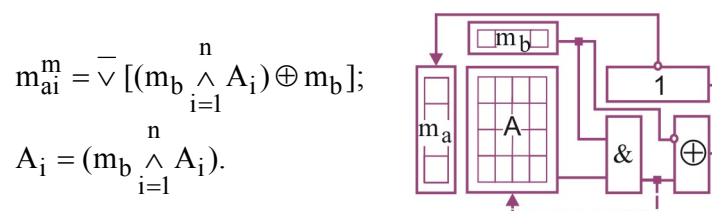


Рисунок 1.8 – Пошук всіх припустимих розв’язків

Можливо й самостійне застосування другої моделі, орієнтоване на визначення однозначного й багатозначного розв'язку при пошуку дефектів у цифровій системі.

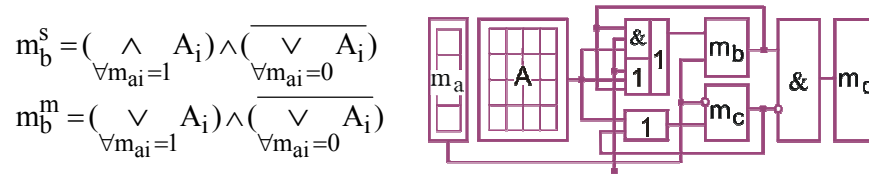


Рисунок 1.9 – Вибір оптимального розв'язку

Всі операції, виконувані двома процес-моделями — векторні. Модель аналізу рядків (див. рис. 1.8) формує вектор m_a ідентифікації припустимих $m_{ai}=1$ або суперечливих $m_{ai}=0$ розв'язків щодо вхідної умови m_b за n тактів обробки всіх m -розрядних векторів таблиці $A = \text{card}(m \times n)$. Якість (допустимість) розв'язку визначається для кожної взаємодії вхідного вектора m_b й рядка $A_i \in A$ на блоці (редукції) диз'юнкції. Матриця A може бути модифікована її перетинанням із вхідним вектором на основі використання операції $A_i = (m_b \bigwedge_{i=1}^n A_i)$, якщо необхідно виключити з A -таблиці всі незначущі для розв'язання координати й вектори, відзначені одиничними значеннями вектора m_a .

Розв'язування задач діагностування за допомогою аналізу рядків таблиці (див. рис. 1.9) здійснюється так. Після виконання діагностичного експерименту формується двійковий вектор експериментальної перевірки m_a , що маскує A -таблицю несправностей для пошуку одиночних або кратних дефектів. Вектори m_b й m_c використовуються для нагромадження результатів виконання операцій кон'юнкції й диз'юнкції. Потім виконується логічне вирахування (xor-операція) з першого регістра m_b вмісту другого вектора m_c з наступним записом результату до регістру m_d . Для реалізації другого рівняння, що формує

множинний розв'язок, елемент *and* замінюється функцією *or*. У схемі використовується також змінна вибору режиму пошуку розв'язку: *single* або *multiple*. Як вхідна умова в моделі використаний вектор m_a , керуючий вибором векторної операції *and*, *or* для обробки одиничних $A_i(m_{ai} = 1) \in A$ або нульових $A_i(m_{ai} = 0) \in A$ рядків *A-таблиці*. У результаті виконання n тактів здійснюється нагромадження одиничних і нульових значень координат вектора m_a рішень у регістрах A_1, A_0 . Априорі в зазначені регістри заноситься вектор одиниць і нулів: $A_1 = 1, A_0 = 0$. Після обробки всіх n рядків *A-таблиці* за n тактів виконується векторна кон'юнкція вмісту регістра A_1 з інверсією регістра A_0 , що формує результат у вигляді вектора m_b , де одиничні значення координат визначають розв'язок. У таблиці несправностей цифрового виробу одиничним координатам вектора m_b відповідають стовпці, що ототожнюються з номерами дефектів або несправних блоків, що підлягають відновленню або ремонту.

При сервісному обслуговуванні функціональних модулів можна на універсальній структурі системи векторного логічного аналізу вирішити оптимізаційне завдання відновлення працездатності. За допомогою мінімального числа ремонтних запасних рядків і (або) стовпців, наприклад пам'яті, необхідно забезпечити квазіоптимальне покриття всіх виявлених в комірках несправностей. Технологічна й математична складові векторної логіки в цьому випадку обумовлюють простий схемотехнічний розв'язок для одержання квазіоптимального покриття (рис. 1.10), переваги якого полягають в наступному: 1. Обчислювальна складність процедури: число векторних операцій дорівнює числу рядків таблиці, $Z = n$. 2. Мінімум апаратних витрат: таблиця й два вектори (m_b, m_a) для зберігання проміжних покриттів і нагромадження результату у вигляді одиничних координат, що відповідають рядкам таблиці, які становлять квазіоптимальне покриття. 3. Відсутність класичного розподілу задач покриття на пошук ядра покриття й доповнення. 4. Відсутність складних процедур маніпулювання комірками рядків і стовпців.

Одержання не завжди оптимального покриття - недолік, що компенсується технологічністю векторної процедури, представленої на рис. 1.10.

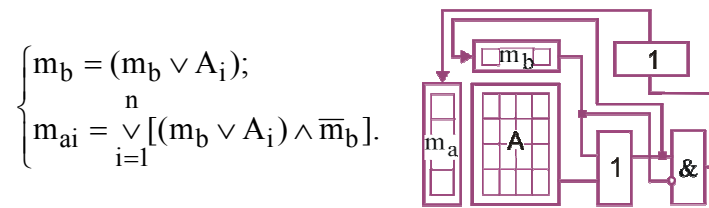


Рисунок 1.10 – Процес-модель пошуку квазіоптимального покриття

Операція редукції, що на останньому етапі перетворює векторний результат у біт m_{ai} вектора m_a за функцією $m_{ai} = \bigvee [(m_b \vee A_i) \wedge \bar{m}_b]$. У загальному випадку операція редукції в алгебрі векторних операцій записується у вигляді <бінарна операція><вектор>: $\bigvee A_i, \wedge m, \bar{\wedge} (m \vee A_i)$. Зворотна процедура – векторизація є конкатенація булевих змінних: $m_a (a, b, c, d, e, f, g, h)$. У процедурі пошуку покриття апріорі вектори $m_b = 0, m_a = 0$ стають рівними. Квазіоптимальне покриття накопичується за n тактів у векторі m_a послідовним зрушенням. Біти, що заносяться в регістр m_a , формуються схемою og , що виконує редукцію після аналізу отриманого результату $[(m_b \vee A_i) \wedge \bar{m}_b]$ на наявність одиниць.

Становить інтерес функціональна закінченість циклу діагностування, коли після одержання квазіоптимального покриття дана інформація використовується для відновлення працездатності дефектних комірок пам'яті. Розмірність модуля пам'яті (13x15 осередків) не впливає на обчислювальну складність одержання покриття десяти дефектних осередків за допомогою резервних рядків і стовпців (рис. 1.11).

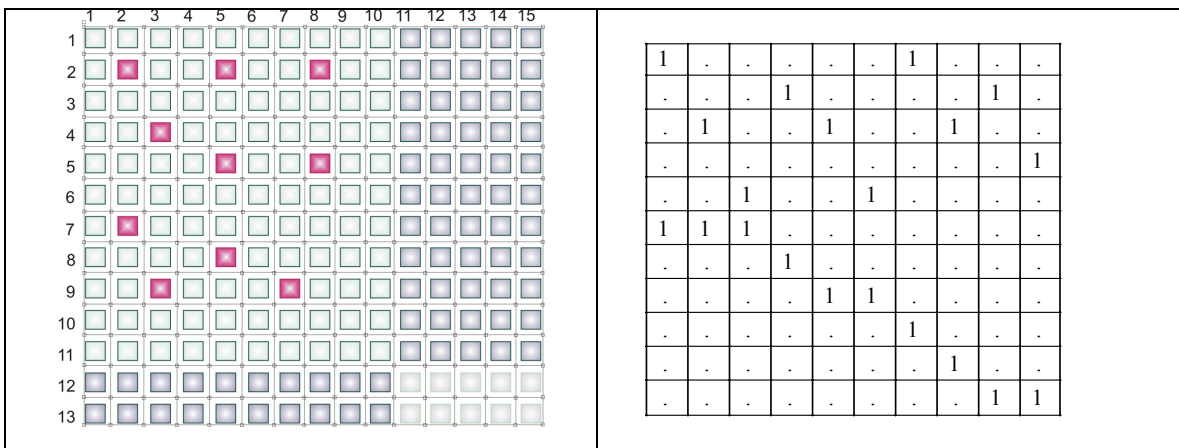


Рисунок 1.11 – Модуль пам'яті з резервом і таблиця покриття

Для розв'язання оптимізаційної задачі виконується побудова таблиці покриття (див. рис. 1.11) несправних осередків, у якій рядки – резервні ресурси для покриття дефектів ($C_2, C_3, C_5, C_7, C_8, C_2, R_2, R_4, R_5, R_7, R_8, R_9$), а стовпці – дефекти осередків ($F_{2,2}, F_{2,5}, F_{2,8}, F_{4,3}, F_{5,5}, F_{5,8}, F_{7,2}, F_{8,5}, F_{9,3}, F_{9,7}$), що підлягають ремонту. При цьому стовпці відповідають координатам дефектних осередків, а рядки ідентифікують резервні компоненти (рядки й стовпці), які можуть відновити працездатність несправних координат. Модель обчислювального процесу, представлена на рис. 1.9, дає можливість одержати оптимальний розв'язок у вигляді $m_a = \boxed{1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0}$, якому відповідає покриття $R = \{C_2, C_3, C_5, C_7, C_8\}$, як одне із трьох можливих мінімальних розв'язків $R = C_2, C_3, C_5, C_7, C_8 \vee C_2, C_3, C_5, C_8, R_9 \vee C_2, C_5, C_8, R_4, R_9$ для таблиці несправностей. Технологічна модель убудованого діагностування й ремонту пам'яті (рис. 1.12) має чотири компоненти:

1. Тестування модуля (Unit Under Test (UUT)) з використанням еталонної моделі (Model Under Test (MUT)) для формування вектора експериментальної перевірки m_a , розмірність якого відповідає числу тестових наборів.
2. Пошук дефектів на основі аналізу таблиці несправностей A .
3. Оптимізація покриття дефектних комірок ремонтними рядками й стовпцями на основі аналізу таблиці A .

4. Відновлення працездатності пам'яті за допомогою заміни адреси (Address Decoder (AD)) несправних рядків і стовпців, представлених вектором m_a , на адреси компонентів з ремонтного запасу (Spare Memory (SM)).

Процес-модель убудованого сервісного обслуговування працює в реальному масштабі часу й дозволяє підтримувати в працездатному стані, без втручання людини, цифрову систему на кристалі, що є доцільним розв'язком у випадку застосування технологій, пов'язаних з дистанційною експлуатацією виробу. Запропоновані процес-моделі аналізу асоціативних таблиць, а також уведені критерії якості логічних розв'язків дозволяють вирішувати задачі квазіоптимального покриття, діагностування дефектів програмних і (або) апаратних блоків. Модель векторних обчислень стала основою для розробки логічної асоціативної мозкоподібної енергозберігаючої мультипроцесорної архітектури, орієнтованої на моделювання і підсилення функціональних можливостей людини, з використанням структур асоціативних таблиць.

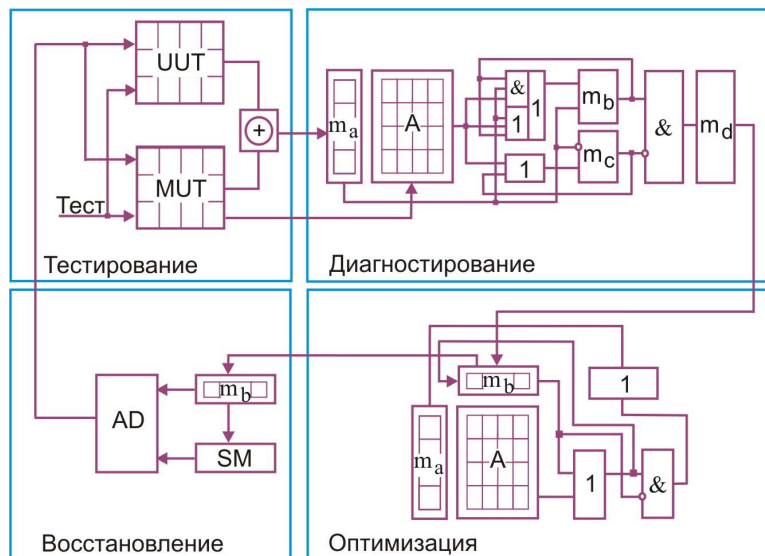


Рисунок 1.12 – Модель убудованого тестування й відновлення пам'яті

Аналітична оцінка ефективності проектного рішення, спрямованого на виконання умов спеціалізації Sp і стандартизації St (рис. 1.13) визначається мінімумом середнього значення наступних трьох взаємно суперечливих

відносних і безрозмірних параметрів: рівень помилок проекту L , час верифікації й (або) тестування T , програмно-апаратна надмірність, обумовлена механізмами асерцій і (або) граничного сканування H .

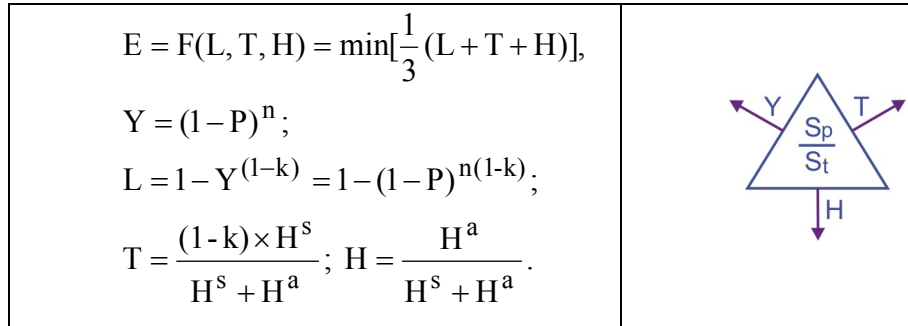


Рисунок 1.13 – Оцінка ефективності процес-моделі

Параметр L , як доповнення до Y , що характеризує вихід придатної продукції, залежить від тестопридатності проекту k , імовірності P існування несправних компонентів і числа невиявлених помилок n . Час верифікації визначається тестопридатністю проекту k , помноженої на структурну складність апаратно-програмної функціональності, віднесеної до загальної складності проекту в рядках коду або еквівалентних вентилів. Програмно-апаратна надмірність перебуває у функціональній залежності від складності асерційного коду або механізму граничного сканування, віднесеної до загальної складності проекту. При цьому асерційна, або скануюча, надмірність повинна забезпечувати задану глибину діагностування помилок функціональності за час виходу виробу на ринок, визначене замовником.

1.6. Висновки до розділу 1

Існуючі програмні аналоги практично не пропонують асоціативно-логічних технологій пошуку, розпізнавання й прийняття рішень у дискретному інформаційному просторі [Васильєв С.Н., Жерлов А.К., Федосов Е.А., Федунов Б.Е. Інтелектуальне керування динамічними системами. – М.: Фізико-математична література. – 2000. – 352 с.; Липаєв В.В. Програмна інженерія.

Методологічні основи. Підручник.– Москва: Теис. – 2006.– 608 с. 15.; Трахтенгерц Э.А. Комп'ютерні методи реалізації економічних і інформаційних управлінських рішень. - СИНТЕГ. - 2009. - 396 с.]. Практично всі вони використовують універсальну систему команд сучасного кошовного процесора з математичним співпроцесором. У той же час, апаратні спеціалізовані засоби логічного аналізу, що є їхніми прототипами [Бондаренко М.Ф. О мозгоподобных ЕВМ / М.Ф. Бондаренко, З.В. Дудар, И.А. Єфімова, В.А. Лещинский, С.Ю. Шабанов-Кушнаренко // Радіоелектроніка й інформатика.– Харків: ХНУРЭ.– 2004, № 2.– С. 89–105; А.С. №1439682. 22.07.88. Регистр сдвига / Какурин Н.Я., Хаханов В.И., Лобода В.Г., Какурина А.Н. – 4с.; Гайдук С.М., Хаханов В.И., Обризан В.И., Каменюка Е.А. Сферичний мультипроцесор PRUS для рішення булевих рівнянь // Радіоелектроніка й інформатика. Харків. 2004. № 4(29). С. 107-116], як правило, орієнтовані на побітову або неекторну обробку інформації.

Запропонований новий підхід векторно-логічної обробки асоціативних даних з повним виключенням арифметичних операцій, що впливають на швидкодію й апаратну складність, може бути ефективно реалізований на основі використання сучасної мікроелектронної апаратури у вигляді мультипроцесорної цифрової системи на кристалі. Фактична реалізація підходу заснована на пропозиції моделей і методів, що використовують загальну ідею векторно-логічної метрики кіберпростору:

1. Процес-моделі аналізу асоціативних таблиць орієнтовані на досягнення високої швидкодії аналізу інформаційних об'єктів і підрахунку критеріїв якості їхньої взаємодії на основі векторних логічних операцій для пошуку, розпізнавання образів, прийняття й оцінювання розв'язків у кіберпросторі.

2. Метод паралельного рішення асоціативно-логічних завдань із мінімальним числом векторних логічних операцій і повним виключенням арифметичних команд, що забезпечує високу швидкодію, мінімальну вартість і

незначне енергоспоживання обчислювача, реалізованого на кристалі програмувальної логіки.

3. Нові векторно-логічні процес-моделі убудованого діагностування цифрових систем на кристалах, пошуку квазіоптимального покриття, що використовують засоби логічного асоціативного мультипроцесора, паралельні операції обчислювальних процесів і векторно-логічний критерій якості.

Практична значимість отриманих результатів підтверджена створенням убудованого компонента для діагностування й відновлення працездатності пам'яті в цифровій системі на кристалі.

Подальші дослідження спрямовані на розробку прототипу логічного асоціативного мультипроцесора для рішення актуальних завдань пошуку, розпізнавання та прийняття рішень за допомогою векторного логічного аналізу.

РОЗДІЛ 2. МЕТОДИ ТА АРХІТЕКТУРНІ ПРИНЦИПИ ШВИДКОДІЮЧИХ АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ ПОПЕРЕДНЬОЇ ОБРОБКИ І КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

2.1. Розробка архітектурних принципів систем обробки інформації в реальному часі на базі нейронної мережі СМАС

Загальні відомості. Нейронні мережі використовуються для моделювання і/або контролю технологічного процесу з метою уникнення математичного моделювання, що пов'язане з такими його особливостями:

1. Математично орієнтоване моделювання вимагає великих витрат інженерної думки для створення опису процесу.

2. Існують процеси, де традиційні способи моделювання не дозволяють отримати результат, тоді як управління людиною таким процесом протікає нормально.

3. Математична модель процесу може бути настільки комплексною, або сам процес настільки швидким, що час обчислень стає критичним.

4. Математична модель прототипу може стати непридатною внаслідок старіння або зміни об'єкта.

Незважаючи на те, що можна використовувати найбільш відомі і популярні нейромережні рішення, такі як мережі з прямим зв'язком, що навчаються за алгоритмом зворотного розповсюдження помилки або його модифікаціями, дослідження показали, що швидкість обчислень і навчання таких мереж дуже мала для застосування в управлінні. Водночас багато дослідників намагалися застосувати спеціалізовані нейронні мережі для вирішення таких завдань.

Нейронна мережа СМАС була запропонована Дж. Альбусом (J. Albus). Абревіатура СМАС означає “Cerebellar Model Articulation Controller” (церебральна модель контролера артикуляції). Мережа так названа, оскільки була моделлю мозочка і використовувалася для управління рукою робота. Теоретичні переваги бачилися у високій збіжності і можливостях модифікації та

удосконалення. Стверджувалося, що мережа СМАС забезпечує високу збіжність до будь-якої комбінації вхід-вихід, що є важливим у завданнях управління і швидкого синтезу функцій. Крім того, прогнозувалася можливість апаратної реалізації мережі СМАС, і, відповідно, отримання високої швидкості навчання і роботи.

Структура простої класичної мережі СМАС наведена на рис. 2.2. Вона може розглядатися як асоціативна пам'ять, яка здійснює два послідовні відображення.

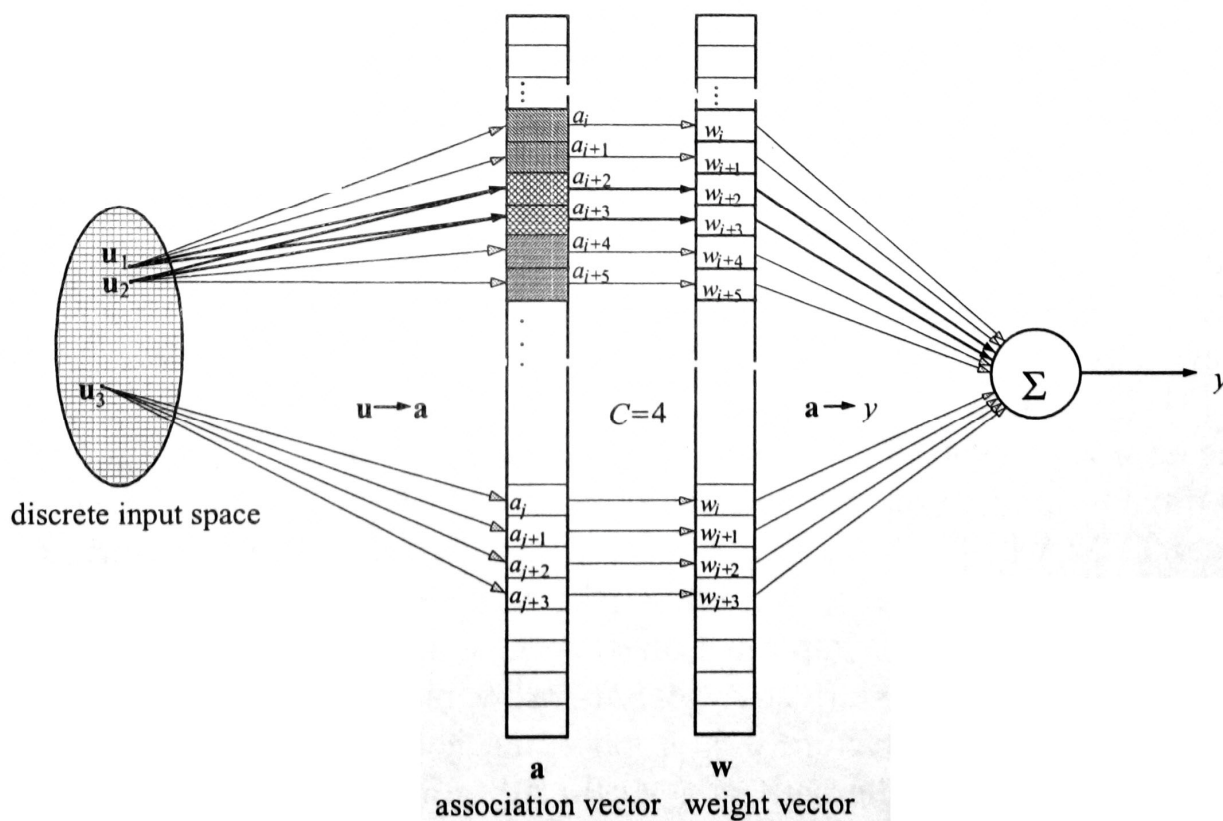


Рисунок 2.1 – Узагальнена структура нейронної мережі СМАС

Хай \mathcal{U} є дискретний простір u вхідних сигналів u_j . Такий простір може утворюватися змінними при синтезі функцій або даними, що отримуються після аналого-цифрового перетворення. Основною особливістю цього простору є кінцевий набір значень. Перше відображення є нелінійним і полягає в тому, що кожному значенню u_j з простору u ставиться у відповідність унікальний двійковий вектор асоціацій \mathbf{a} , що містить ρ одиничних біт. Решта біт вектора асоціацій, відповідного даному вхідному значенню, рівні нулю. Слід зазначити,

що значення ρ значно менше довжини вектора асоціацій. Цей параметр визначає узагальнювальні властивості мережі і часто називається узагальнювальним параметром. Кожному біту вектора асоціацій ставиться у відповідність двійкова базисна функція, яка рівна одиниці для ρ вхідних сигналів. З точки зору математики вона може задаватися так:

$$f_n(u) = \begin{cases} 1, & u \in [u_k, u_{k+1}, u_{k+2}, \dots, u_{k+\rho-1}] \\ 0, & u \notin [u_k, u_{k+1}, u_{k+2}, \dots, u_{k+\rho-1}] \end{cases} \quad (2.1)$$

У наведеній залежності u_k і u_{k+1} відрізняються на величину кроку квантування вхідних сигналів, який при двійковому поданні сигналів рівний одиниці молодшого розряду. Набір значень вхідних сигналів, на яких базисна функція рівна 1, називається її рецептивним полем.

Відображення вхідного простору у вектор асоціацій має такі властивості:

2. Два сусідні вхідні значення відображуються у вектори асоціацій, що відрізняються лише декількома елементами.

2. Із зростанням кодової відстані між вхідними значеннями кількість загальних активних бітів у векторі асоціацій зменшується. Якщо це відстань більша ρ , вектори асоціацій не повинні мати загальних одиничних бітів.

Друге відображення обчислює вихід нейронної мережі як скалярний твір вектора асоціацій a на вектор ваг w . З урахуванням того, що вектор асоціацій є двійковим,

$$y(u) = \sum_{i:a_i(u)=1} w_i \quad (2.2)$$

Так, проста нейронна мережа СМАС для своєї реалізації не вимагає виконання ресурсомістких і таких, що вимагають часових витрат, операцій добутку, що робить привабливою її апаратну реалізацію.

Далі детальніше розглянемо наведену структуру, покажемо можливість вибору значень базисних функцій, відмінних від одиничних, вкажемо шляхи скорочення довжини вектора вагових коефіцієнтів, розглянемо навчання мережі. У завершенні розділу будуть розглянуті можливості застосування нейронної мережі СМАС для обробки зображень в реальному часі.

2.1.1 Етапи обробки інформації

Кодування інформації. Різке зменшення об'єму пам'яті, необхідної для зберігання оброблюваної інформації, є особливістю мережі СМАС. Ця особливість реалізується за рахунок застосування в мережі спеціального кодування.

Кодування інформації здійснюється в шарах L1 і L2 і полягає в тому, що кожному N-вимірному вхідному вектору $x(i)$ ставиться у відповідність n-вимірний вектор асоціацій $a(i)$. При цьому лише $\rho \ll n$ елементів даного вектора мають відмінні від нуля значення. Іншими словами, для обчислення функції, відповідної вхідному вектору $x(i)$, необхідно прочитати вміст ρ елементів пам'яті, що зберігає значення ваг, що настроюються. Таку ж кількість ваг необхідно обчислити і записати в пам'ять при навчанні мережі на вхідному векторі $x(i)$.

Кодування здійснюється в два етапи. На першому етапі відбувається дискретизація безперервної безлічі вхідних сигналів (квантування за рівнем), після чого отримана дискретна множина аналізується нейронами вхідного шару на предмет входження в область визначення їх функцій активації. При цьому на кожному ступені квантування активізується один нейрон для кожного елемента вектора $x(i)$. Якщо вхідні сигнали дискретні за своєю природою (наприклад, координати точок, яскравість яких обчислюється мережею), операція дискретизації втрачає необхідність. На другому етапі кодування асоціативні нейрони другого шару аналізують, які нейрони першого шару активні і формують вектор асоціацій. Схема кодування інформації однозначно визначається розподілом нейронів вхідного шару за ступенями квантування.

Розглянемо причину скорочення об'єму інформації. За відсутності кодування на вхідному шарі кожному рівню квантування однієї з вхідних змінних необхідно зіставити всі можливі значення інших змінних. Для кожного такого зіставлення необхідно зберігати значення функції. Кількість значень рівна добутку кількостей рівнів квантування вхідних змінних. Під час використання кодування зіставляються не рівні квантування, а області. При цьому на кожній області на j -му ступені кодування однієї змінної зіставляються всі області квантування інших змінних, що належать лише j -му ступеню. Кількість областей квантування на ступені визначається відношенням кількості рівнів квантування R змінної до рівнів квантування, що об'єднуються областю. Ступінь може містити до двох неповних областей на краях, поява яких пов'язана із зміщенням розподілу областей на сусідніх ступенях. Ці міркування пояснюють суть залежності, що визначає максимальну кількість зіставлень при даних параметрах вхідних змінних і самої мережі.

Отже, значення ваги, що зберігається в даному елементі асоціативної пам'яті мережі СМАС, використовується при різних комбінаціях вхідних сигналів (близьких в плані відстані Хеммінга), тоді як за відсутності кодування один елемент пам'яті відповідає єдиній комбінації вхідних змінних. Таке кодування припускає, що близькі (у плані відстані Хеммінга) вхідні змінні мають корельовані відповідні значення вихідних функцій. Це твердження справедливе для яскравостей і кольорів сусідніх точок на повнокольорових зображеннях, для функцій присутності, використовуваних при мікшуванні зображень, для функцій колірної корекції. В результаті цього мережу СМАС можна запропонувати для стиснення зображень, для синтезу функцій присутності і колірної корекції.

Для оцінки скорочення об'єму необхідної пам'яті для мережі СМАС у порівнянні з табличним методом розглянемо приклад на рис. 2.2. Для табличного способу кодування необхідне $R^2 = 19^2 = 361$ елементів пам'яті. Водночас для мережі СМАС – максимум 106 комірок, тобто більш ніж в три рази менше. Для

випадку 3-х вхідних змінних, що мають розрядність 8 біт ($R = 256$), при $\rho = 8$ економія складе:

$$\left\lceil \frac{256^3}{8 \cdot \left(\frac{256-1}{8} + 1\right)^3} \right\rceil = 59 \text{ раз.}$$

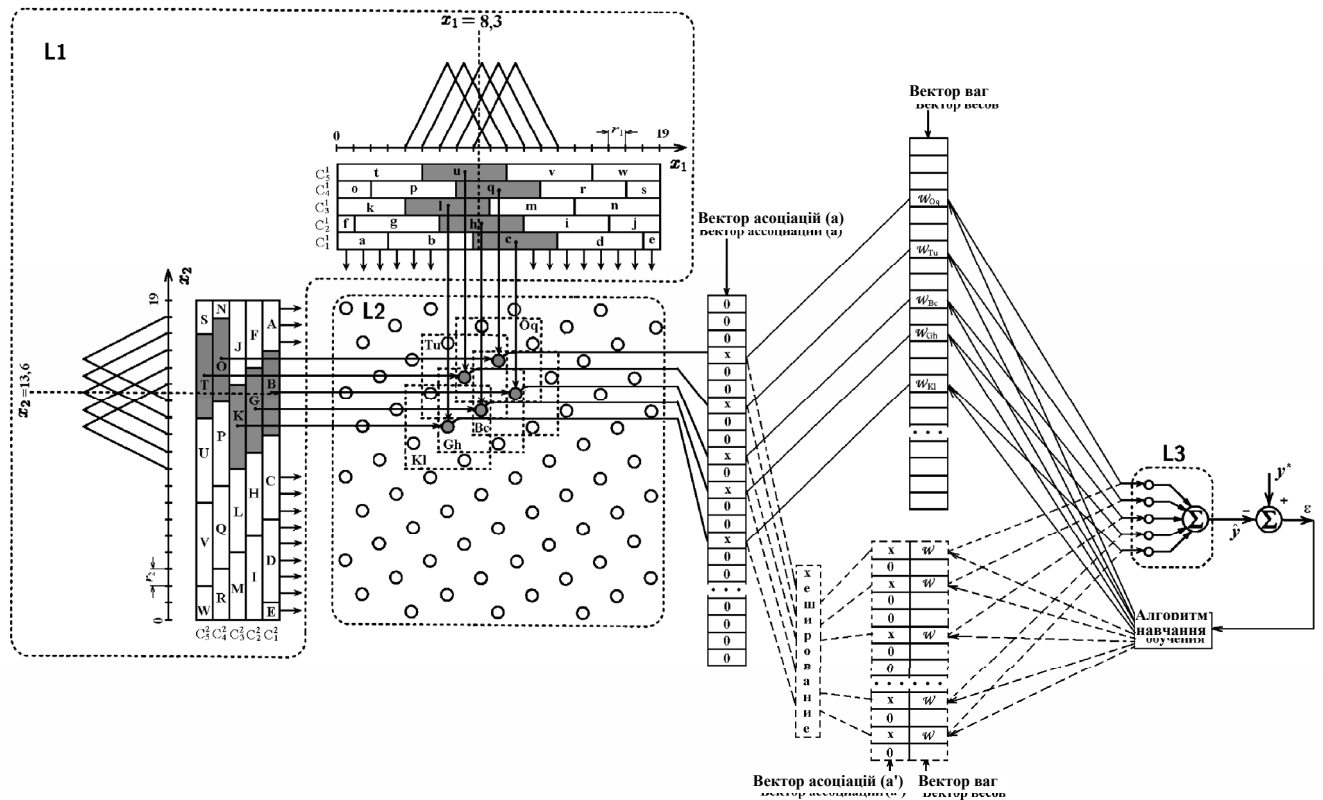


Рисунок 2.2 – Структура нейронної мережі СМАС

Останній приклад показує економію пам'яті під час використання для колірної корекції зображення, наведеного в колориметричній системі RGB, мережі СМАС у порівнянні з табличним методом.

Подальше скорочення об'єму пам'яті можливе під час використання хешування вектора асоціацій.

Зрушення областей на ступенях квантування обумовлюється тим, що лише $(\rho - 1)$ комірок вектора асоціацій можуть співпадати при кодуванні різних вхідних сигналів. На сусідніх ступенях області квантування можуть

змищуватися як на один рівень, так і на декілька. Величини зрушень для кожного ступеня і змінної визначають схему кодування у вхідному шарі.

Вибір базисних функцій. Як уже зазначалося, класична мережа СМАС використовує нейрони вхідного шару з прямокутною функцією активації і, відповідно, виконує кусково-лінійну апроксимацію. Проте, як показують дослідження, на апроксимуючі властивості мережі істотно впливає вибір цих функцій. На рис. 2.3 наведено різні види базисних функцій нейронів першого шару.

При виборі базисних функцій прямокутної форми (рис. 2.3,а) обчислювальні витрати будуть мінімальними. Це обумовлено тим, що у вхідному шарі замість обчислень значень базисних функцій для кожного вхідного сигналу достатньо визначити, які нейрони збуджені, і подати на їхні виходи сигнал, що дорівнює 2. У шарі асоціативних нейронів також не відбувається обчислень, а визначаються збуджені нейрони і з їх виходів знімається сигнал “1”. Внаслідок цього значно скорочується час реакції мережі на вхідний сигнал, що поступив. Компоненти вектора асоціацій в цьому випадку можуть приймати значення “0” або “1”. Швидкість навчання мережі при виборі базисних функцій прямокутної форми буде максимальною. Істотним також є той факт, що в мережі СМАС з базисними функціями прямокутної форми схема кодування інформації не впливає на точність апроксимації, тобто будь-яка схема кодування є прийнятною.

На рис. 2.3 показано активаційні функції збуджених нейронів s, h, l, q, u (див. приклад, наведений на рис. 2.2), що беруть участь у кодуванні координати $x = (0,83 \ 1,36)^T$, подані у вигляді В-сплайнів першого (рис. 2.3,а), другого (рис. 2.3,б) і четвертого (рис. 2.3,в) порядків. Безперечною перевагою В-сплайнів є можливість рекурентного обчислення як самих В-сплайнів, так і їх похідних δ -го порядку.

На рис. 2.3,г показано активаційні функції Гауса цих же збуджених нейронів, що мають вигляд:

$$\Phi_i(x_j) = \exp\left\{-\frac{(x_j - \mu_i)^2}{\sigma^2}\right\}. \quad (2.3)$$

Легко побачити, що вони також допускають достатньо просте обчислення похідних $\frac{\partial \Phi_i}{\partial x_j}$; $\frac{\partial^2 \Phi_i}{\partial x_j^2}$

Проте, хоча функція Гауса і має властивість локального збудження, достатньо складно чітко виділити її межі збудження, що є важливим при реалізації кодування інформації в СМАС. Для усунення цього недоліку можна використати модифіковану функцію Гауса, яка має вигляд:

$$\Phi_i(x) = \begin{cases} \exp\left\{-\frac{(\lambda_2 - \lambda_1)^2 / 4}{(x - \lambda_1)(\lambda_2 - x)}\right\} & \text{при } x \in (\lambda_1, \lambda_2); \\ 0 & \text{інакше.} \end{cases} \quad (2.4)$$

Як видно з (2.4), дана функція точно визначена в інтервалі (λ_1, λ_2) , що спрощує процедуру масштабування базисних функцій при зміні таких параметрів мережі, як R і ρ . На рис. 2.3,д показано модифіковані активаційні функції Гауса збуджених нейронів s, h, l, q, u .

Аналогічний вигляд матимуть і базисні функції збуджених нейронів T, O, K, G, B , що беруть участь в кодуванні другої координати x_2 (у нашому прикладі $x_2 = 1,36$). Оскільки значення компонент вектора асоціацій, що з'являються на виході $L2$, отримується перемножуванням відповідних значень базисних функцій збуджених нейронів.

Як видно з наведених рисунків, дані базисні функції мають поблизу межі області квантування (зони чутливості нейрона вхідного шару) порівняно невеликі значення, що негативно позначається на результатах апроксимації.

Для уникнення вказаного недоліку як базисні можливе використання тригонометричних функцій, наприклад, косинусоїдальної, що має вигляд:

$$\hat{O}_i(x_j) = \begin{cases} \cos\left(\frac{\pi}{\rho r_j}(x_j - \lambda_i)\right) & \text{і де } \delta_j \in (\lambda_i - \frac{\rho r_j}{2}, \lambda_i + \frac{\rho r_j}{2}]; \\ 0 & \text{інакше} \end{cases} \quad (2.5)$$

де λ_i – i -й центр області квантування; r_j – крок квантування після j -ї компоненти вхідного сигналу.

Вид даної функції наведено на рис. 2.3,е. Тригонометричні функції також дозволяють легко обчислювати їх похідні. З рисунка видно, що мінімальні значення базисних функцій збільшилися.

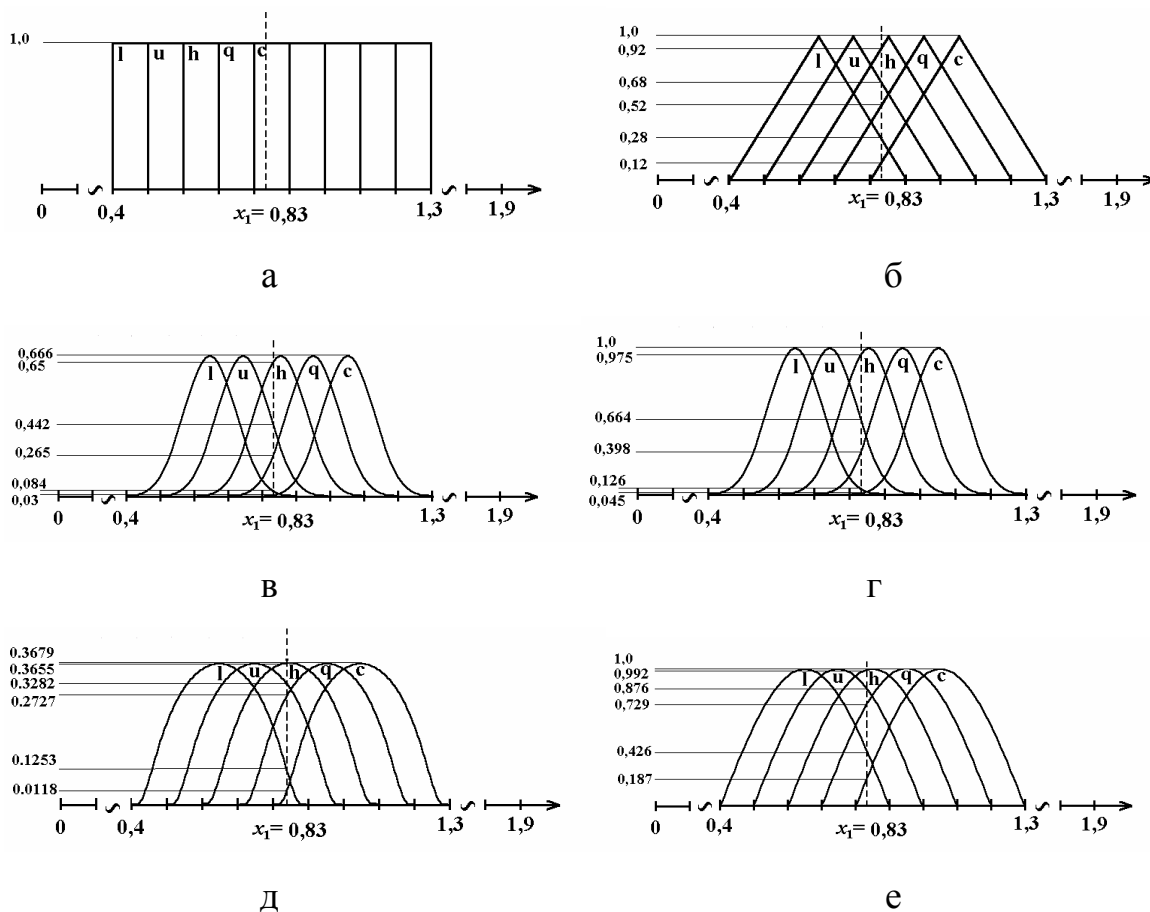


Рисунок 2.3 – Різні види базисних функцій

Як було зазначено вище, ненульові компоненти вектора асоціації утворюються шляхом перемножування відповідних значень базисних функцій збуджених нейронів шару L2. На рис. 2.4 показано, як здійснюється добуток відповідних базисних функцій, і який вигляд має функція, що є результатом такого добутку.

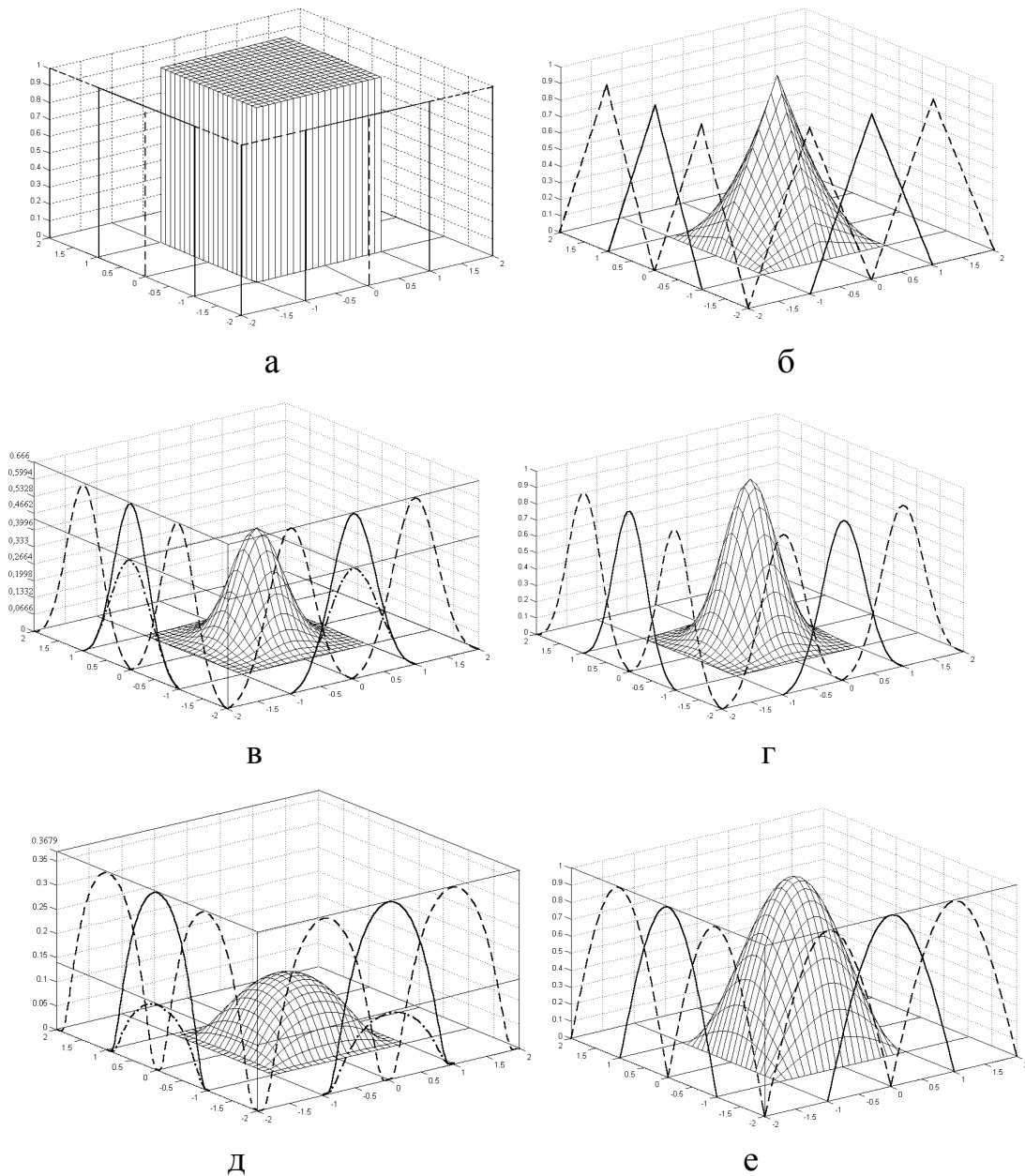


Рисунок 2.4 – Результати перемножування базисних функцій різного вигляду

З рис. 2.4 видно, що при виборі базисних функцій вигляду, відмінного від прямокутного, ці функції визначають, в яку зону області, утвореної базисними функціями нейронів першого шару, виходи яких поступають на один і той самий асоціативний нейрон другого шару, входить даний вхідний вектор. Чим ближче до краю області даний вхідний сигнал, тим менше значення має добуток базисних функцій, і тим менший вплив надаватиме вага, відповідно даному асоціативному нейрону на вихідний сигнал мережі. Під час навчання мережі ця вага буде найменш змінена.

Хешування інформації. Для скорочення об'єму пам'яті, необхідного для зберігання ваг, можливе застосування хешування інформації, яке полягає у відображенні більшої, необхіднішої пам'яті в меншу, яка простіше реалізується фізично і швидше ініціалізується. Найпростішим прикладом хешування інформації є обчислення залишку від ділення великого числа на деяке відоме, відносно коротке число. В цьому випадку, яким би не було перше число, результат його хешування буде не більш, ніж модуль (число, на яке проводиться ділення). Очевидно, що в результаті хешування може виникнути ситуація, коли різні початкові дані призводять до одного результату – виникає так звана хеш-колізія, що носить імовірнісний характер. Вірогідність хеш-колізії залежить від співвідношення довжин інформаційного поля і модуля.

У мережі СМАС початковою для операції хешування є адреса ваги в пам'яті, що отримується з вектора асоціацій і що лежить в інтервалі $[0, M-1]$. Результат – адреса ваги у фізичній пам'яті, що лежить в інтервалі $[0, m-1]$, де $m < M$. Функція перетворення, або хеш-функція, має відповідати вимозі:

$$0 \leq H(k) \leq m - 1, \forall k : 0 \leq k \leq M - 1.$$

Вибір хеш-функції $H(k)$ є дуже трудомістким завданням, проте численні дослідження показали, що хороші результати забезпечує застосування двох

основних типів хеш-функцій, один з яких заснований на діленні, а інший – на множенні.

Перший випадок відповідає розглянутому прикладу хешування і аналітично може бути записаний у вигляді

$$H(k) = k \bmod m, \quad (2.6)$$

де функція \bmod означає ділення за модулем, відповідне обчисленню залишку від цілочисельного ділення k на m . Така функція рівномірно розподіляє велику пам'ять в меншу.

У другому випадку хеш-функція обчислюється за формулою

$$H(k) = m \left[\text{frac} \left(\frac{F}{w} k \right) \right], \quad (2.7)$$

де функція frac означає узяття дробової частини виразу, F – деяка ціла константа, взаємно проста з розміром машинного слова w , $\lceil \cdot \rceil$ – означає округлення у бік найближчого більшого цілого числа. Як F можна використовувати числа Фібоначчі. Хеш-функція, заснована на множенні, «тасує» елементи пам'яті.

Розглянемо вплив застосування алгоритмів хешування на результат апроксимації на прикладі функції «Виключне АБО». На рис. 2.5, а наведений результат реалізації даної функції за допомогою класичної мережі СМАС із такими параметрами: $R = 50$; $\rho = 20$; $n = 236$. На рис.2.5, б–д наведено реалізацію функції «Виключне АБО» із застосуванням різних алгоритмів хешування.

Як вже зазначалося, незалежно від використовуваного методу хешування, повністю усунути хеш-колізії неможливо. Колізії першого типу виникають при активації одним вхідним сигналом одного елемента пам'яті більше одного разу. Вірогідність того, що один вхідний сигнал звернеться два і більше разів до одного елемента фізичної пам'яті, можна визначити за формулою

$$P = \sum_{i=2}^{\rho-1} \frac{i}{m}. \quad (2.8)$$

Причиною виникнення колізій другого типу є активація одного й того ж елемента пам'яті різними далеко розташованими один від одного (у значенні відстані Хеммінга) вхідними сигналами. Ці колізії призводять до серйознішої проблеми – появи небажаної кореляції між двома далеко розташованими (у значенні відстані Хеммінга) вхідними сигналами. Ці колізії не можуть істотно вплинути на роботу мережі, поки виконується умова

$$a'_1 \wedge a'_2 \ll \rho, \quad (2.9)$$

де a'_1 і a'_2 – перетворені хеш-функцією вектори асоціацій двох різних вхідних сигналів.

Виникнення хеш-колізій призводить до того, що на одну і ту ж вагу впливають далекі в значенні відстані Хеммінга сигнали. Для таких сигналів значення обчислюваної функції слабо корельовані або не корельовані взагалі. Проте хеш-колізія внесе кореляцію і додасть шум до вихідного сигналу мережі.

У плані апаратної реалізації найбільш очевидний і простий алгоритм хешування (2.6). Оскільки використовувана пам'ять має об'єм рівний, як правило, степені числа 2, операція ділення за модулем зводиться до взяття як результат обчислення g – молодші розряди початкової адреси ваги ($m = 2^g$).

2.1.3 Методи навчання

Навчання мережі СМАС, як практично і всіх інших ШНС, полягає в настроюванні вектора її вагових параметрів w розмірності $n \times 1$, тобто в даній мережі настроюються лише ваги зв'язків між асоціативними нейронами і нейронами вихідного шару. Використання спеціального кодування дозволяє зберігати в ρ елементах пам'яті ваги мережі. Якби різні вхідні вектори при кодуванні не мали загальних комірок, то ваги мережі могли визначатися і без її навчання, аналітично, так:

$$w_{ik} = \frac{y(k)}{\rho}. \quad (2.10)$$

Наявність загальних областей пам'яті для зберігання різних векторів не дозволяє обмежитися таким простим обчисленням ваг і обумовлює необхідність навчання мережі. Компоненти вектора ваг мають фізичні адреси і вибираються так, щоб середнє арифметичне значення активних ваг, тобто ваг, відповідних одиничним компонентам вектора асоціацій, було рівне значенню функції, що запам'ятовувалася, $y(i)$.

Наявність загальних елементів пам'яті для різних векторів призводить до колізій, тобто явища, при якому мережа обчислює для двох різних вхідних векторів однакові адреси.

Алгоритм навчання Альбуса. Під час навчання мережі подають навчальні $(x(i), y^*(i))$, $i = \overline{1, M}$, де $y^*(i)$ – необхідне значення функції, і коректуються значення вагових параметрів так, щоб із зростанням кількості подавань ця послідовність ваг, що настроюються, прагнула до стаціонарного значення w^* .

Вихідний сигнал мережі на k -му такті навчання можна записати так:

$$y(k) = wT(k)a(k), \quad (2.11)$$

де $w(k) = (w_1(k), w_2(k), \dots, w_n(k))^T$ – вектор оцінок вагових параметрів.

Як вже наголошувалося, ваги $w(k)$ вибираються так, щоб їх середнє арифметичне було рівне значенню функції, що запам'ятовувалася. А оскільки на кожному такті навчання (при поданні вхідного вектора) тільки ρ компонент вектора асоціацій a відмінні від нуля (активними виявляються ρ елементів пам'яті $\rho \ll n$), то вираз (2.11) може подаватися так:

$$y(k) = \frac{1}{\rho} \sum_{i=1}^n w_i(k), \quad (2.12)$$

де підсумовування відбувається по всіх i , для яких $a_i(k) = 1$.

З урахуванням цього запропоновано рекурентний алгоритм навчання СМАС, згідно з яким при поданні чергового вхідного сигналу коректуються лише ті ρ ваг мережі, які відповідають одиничним компонентам вектора асоціацій для даного сигналу. Аналітично алгоритм навчання Альбуса можна записати у такій формі:

$$w_i(k) = \begin{cases} w_i(k-1) + \left(y(k) - \sum_i w_i(k-1) \rho^{-1} \right) \\ \forall i, a_i(k) = 1, a_i(k-1) = 1; \forall j, a_j(k) = 1, \\ w_i(k) = w_i(k-1) \quad \forall i, a_i(k) = 1. \end{cases} \quad (2.13)$$

Отже, корекція вагового вектора $w(k)$, обчисленого за k вимірювань, при набутті $k+1$ -го значення вхідного вектора $x(k+1)$ і відповідного значення функції $y^*(k+1)$ відбувається за таким правилом: за вхідним вектором $x(k)$ шляхом кодування визначається ρ -вимірний вектор асоціацій $a(k)$, визначаються відповідні активні асоціативні елементи вектора ваг $w(k)$; із значення функції $y^*(k)$, що запам'ятовується, віднімається середнє арифметичне значення активних ваг вектора $a(k-1)$ (оскільки всі ваги при виборі прямокутних активаційних функцій є рівноправними), а отримана різниця

додається до значень активних ваг; неактивні ж значення вагового вектора не коректуються.

Метою алгоритму Альбуса є визначення вектора w , який прагне до постійного вектора w^* , такого, як

$$\hat{a}_i w^* = \sqrt{\rho} y_i^* \quad (2.14)$$

для кожного i ($i=1, 2, \dots, N$). Оскільки вектор w^* містить ρ елементів, (2.14) подає N рівнянь з ρ невідомими й існування розв'язання (або розв'язань) цієї системи визначається відомими результатами з лінійної алгебри. Запишемо (2.14) в матричній формі

$$Aw^* = h, \quad (2.15)$$

де i -й рядок матриці A розміру $N \times \rho$ складається з елементів ρ -вимірного вектора $\hat{a}_{(i)}^T$ (одична нормаль до гіперплощини, вибрана на i -му кроці), а $(h - N)$ -вимірний вектор-стовпець, i -й елемент якого рівний $\sqrt{\rho} y^*(i)$. Щоб система рівнянь (2.14) була сумісна, матриця A і розширена матриця $[A, h]$ повинні мати однаковий ранг, рівний, наприклад R . Для існування єдиної точки x^* потрібно, щоб $N \geq \rho$ і, більш того, ранг R має досягати свого найбільшого значення, тобто $R = \rho$. Якщо $N > \rho$, то $N - \rho$ надмірних рівнянь можуть виключатися з системи, то ρ рівнянь, що залишилися, записуються у вигляді $A^1 w^* = h^1$, де A^1 — матриця розміру $\rho \times \rho$, ранг якої рівний ρ , а h^1 — ρ -вимірний вектор-стовпець, який дозволяє при відомій зворотній матриці A^1 однозначно визначити вектор $w^* = (A^1)^{-1} h^1$.

Існують й інші можливості під час розв'язання системи рівнянь, наприклад у тому випадку, коли система рівнянь (2.15) несумісна. Ці можливі

ситуації наведені в табл. 2.1, де вказаний їх вплив на поведінку алгоритму Альбуса (2.13). Наведені в табл. 2.1 випадки 3, 5 і 7, коли система рівнянь несумісна, можуть зустрічатися на практиці.

Зазначимо, що розв'язання системи рівнянь (2.15) має вигляд

$$w^* = A^+ h^1, \quad (2.16)$$

де A^+ – псевдо звернена матриця за Муром-Пенроузом, визначена виразами

$$A^+ = \begin{cases} (A^T A)^{-1} A^T & \text{при } N \geq \rho; \\ A^T (A A^T)^{-1} & \text{при } N < \rho. \end{cases} \quad (5.17)$$

$$(5.18)$$

У тому випадку, коли система рівнянь несумісна (випадки 5 і 7), це розв'язання, крім того, має таку бажану властивість, як розв'язання з мінімальною нормою. Основною проблемою, що виникає під час реалізації даного розв'язання, є істотна обчислювальна складність, обумовлена великою розмірністю матриць A $N \times p$, які на практиці складаються з тисяч рядків і стовпців, так що обчислення відповідно до формул (2.17), (2.18) виявляються фактично неможливими. Тому в обчислювальному відношенні алгоритм Альбуса, що використовує тільки обчислення скалярного твору два p -вимірних векторів, причому один з цих векторів \hat{a}_i^T містить всього ρ ($\rho \ll p$) відмінних від нуля елементів, має величезну перевагу.

Оскільки на кожному такті навчання $\|a(k)\|^2 = \sum_{i=1}^n a_i^2(k) = \rho$, алгоритм

(2.13) можна записати в еквівалентній векторній формі:

$$w(k+1) = w(k) + \frac{\rho y^*(k) - w^T(k) a(k)}{\|a(k)\|^2} a(k), \quad (2.19)$$

що є алгоритмом Качмажа, і відомий в теорії ШНС як алгоритм Уїдроу і Хоффа. Під час використання алгоритму на кожному такті навчання всі ρ коректованих ваг змінюються на однакову величину.

Таблиця 2.1 – Різні варіанти розв’язання рівняння $Aw^* = h$ (2.15) і їх вплив на алгоритм Альбуса

Випадок	Співвідношення між розмірами p і N	Ранг матриці	Ранг розширеної матриці $[A, h]$	Перетин гіперплощин, визначених (2.14)	Властивості збіжності алгоритму	
					циклічне навчання	випадкове навчання
1	$p \geq N$	N (максимальн 0- можливий)	N	Перетином ϵ гіперплощина розмірності $p - N$	Збіжність до єдиної точки, якщо початкова точка x_0 лінійно залежить від \bar{a}_i , інакше – до точки, яка залежить від x_0	
2	$p \geq N$	$R < N$	$R < N$	Перетином ϵ гіперплощина розмірності $p - R$	Як і у випадку 1	
3	$p \geq N$	$R < N$	$R + 1 < N$	Рівняння (2.14) несумісні. Не існує єдиної гіперплощини перетину	Граничний цикл існує, його форма залежить від порядку циклічного навчання, але не залежить від x_0 . Розташування не залежить від x_0 , якщо x_0 лінійно залежить від \bar{a}_i	Точки, що породжуються алгоритмом (2.15), лежать в мінімальній зоні захоплення з вірогідністю 1
4	$N \geq p$	p (максимальн)	p	Перетином ϵ гіперплощина	Збіжність в цю єдину точку з будь-яких початкових умов x_0	

			розмірності 0, тобто єдина точка		Точки, що породжуються алгоритмом (2.15), лежать в мінімальній зоні захоплення з вірогідністю 1
5	$N \geq p$	p (максимальн о- можливий)	$p + 1$	Рівняння (2.14) несумісні. Не існує єдиної гіперплощини перетину	Граничний цикл існує, залежить від x_0, x_0 порядку навчання, але не залежить від x_0, x_0
6	$N \geq p$	$R < p$	$R < p$	Перетином є гіперплощина розмірності $p - R$	Збіжність до єдиної точки, якщо початкова точка x_0 лінійно залежить від \bar{a}_i , інакше – до точки, яка залежить від x_0
7	$N \geq p$	$R < p$	$R + 1 \leq p$	Рівняння (2.14) несумісні. Не існує єдиної гіперплощини перетину	Граничний цикл існує, як і у випадку 3
					Як у випадку 5

Слід зазначити, що алгоритм Качмажа, у свою чергу, є окремим випадком (2.18), оскільки в ньому використовується операція псевдо звернення вектора $a(k)$.

На відміну від алгоритму Альбуса, в алгоритмі Качмажа елементи вектора можуть бути довільними дійсними числами, тоді як елементи вектора $a(k)$ в алгоритмі Альбуса – бінарні числа. Отже, алгоритм навчання Альбуса – окремий випадок алгоритму Качмажа, і, отже, до алгоритму Альбуса застосовні всі відомі до теперішнього часу висновки щодо алгоритму Качмажа.

Властивість збіжності послідовності $w(k)$ досліджені в роботах Мілітцера і Паркса. У цих роботах було вивчено два способи навчання: циклічне навчання і випадкове.

Циклічне навчання – це така форма навчання мережі, при якій процес обходу N точок навчання повторюється в заданому порядку. При невеликих значеннях N можна отримати матричне співвідношення, яке пов'язує вагові вектори на початку і кінці циклу, та досліджувати збіжність алгоритму за поведінкою власних значень відповідної квадратної матриці.

Під час дослідження питань збіжності алгоритму (2.13) використовувалася функція Ляпунова

$$V(k) = y^T(k-1)L(I_N + L + L^T)^{-1}L^T y(k-1), \quad (2.20)$$

$$\text{де } L = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \hat{a}_2^T \hat{a}_1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 0 \\ \hat{a}_N^T \hat{a}_1 & \hat{a}_N^T \hat{a}_{N-1} & \dots & 0 \end{bmatrix}, \quad I_N - \text{одинична матриця розмірності } N \times N.$$

Як впливає з результатів цих робіт, необхідною і достатньою умовою збіжності алгоритму (2.13) є позитивна визначеність матриці $(I_N + L^T)(I_N + L + L^T)(I_N + L)$ або, що еквівалентно, матриці $(I_N + L + L^T)$.

Оскільки

$$(I_N + L + L^T) = \begin{bmatrix} 1 & \hat{a}_1^T \hat{a}_2 & \dots & \hat{a}_1^T \hat{a}_N \\ \hat{a}_2^T \hat{a}_1 & 1 & \dots & \hat{a}_2^T \hat{a}_N \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \hat{a}_{N-1}^T \hat{a}_n \\ \hat{a}_N^T \hat{a}_1 & \hat{a}_N^T \hat{a}_2 & \dots & 1 \end{bmatrix}$$

є матрицею Грама, відповідно властивості цієї матриці необхідна і достатня умова збіжності, яка полягає в тому, щоб одиничні нормальні вектори \hat{a}_i ($i=1, 2, \dots, N$) були лінійно незалежні. Отже, необхідно, щоб $p \gg N$ і щоб ранг матриці A дорівнював N . Відповідає випадку 1 з табл. 2.2.

При випадковому навчанні на відміну від заданого циклічного порядку, розглянутого вище, кроки навчання здійснюються випадково з множини цілих чисел $1, 2, \dots, N^1$.

Вибираючи функцію Ляпунова вигляду

$$V(k+1) = (w(k+1) - w^*)^T (w(k+1) - w^*), \quad (2.21)$$

можна показати, що якщо одиничні вектори $\hat{a}_1, \hat{a}_2, \dots, \hat{a}_N$ лінійно незалежні,

$$\begin{aligned} \Delta V(k+1) &= V(k+1) - V(k) = (w(k+1) - w^*)^T (w(k+1) - w^*) - \\ &\quad - (w(k) - w^*)^T (w(k) - w^*) \leq 0. \end{aligned}$$

Тут $\mu(k+1) = \sqrt{\rho} y^*(k+1) - a^T(k+1)w(k)$.

Тобто алгоритм (2.13) збігається до фіксованої точки

$$w^* = A^T (I_N + L + L^T)^{-1} Y(k), \quad (2.22)$$

що знаходиться на перетині всіх N^1 гіперплощин.

Це підтверджує той факт, що коли на $k+1$ -му кроці опускається псевдоперпендикуляр з точки $w(k)$ в точку $w(k+1)$, лежачи на гіперплощині $a^T(k)w = \sqrt{\rho}y^*(k)$, то в прямокутному трикутнику ABC з верхівками $A = w^*$, $B = w(k)$, $C = w(k+1)$, гіпотенуза AB довша за катет AC , прилеглий до прямого кута ACB (див. рис. 2.6,а). Отже, алгоритм породжуватиме послідовність точок $w(k)$, які прагнутимуть до w^* при $k \rightarrow \infty$ (цей висновок є правильним і для циклічного навчання).

Фіксована точка w^* існує і у разі, коли одиничні нормальні вектори не є лінійно незалежними, а система рівнянь (2.15) сумісна. Для несумісної системи рівнянь (2.15) не існує такої точки w^* , яка була отримана вище, і при випадковому навчанні не можна навести числові приклади, які можна було зробити у разі циклічного навчання. Тому були висловлені тільки деякі припущення, які були підтверджені для невеликих значень N^1 і які, мабуть, справедливі і в більш загальному випадку.

У даній роботі використовується поняття зон захоплення N^1 наборів точок, кожна з яких лежить в одній з N^1 гіперплощин, визначених рівнянням $Ax=h$ так, що кожен набір точок ортогонально проектується на інші N^1-1 набори точок. Мінімальна зона захоплення – така зона захоплення, яка не може бути далі зменшена в розмірах без втрати властивостей ортогональності зони захоплення.

Тепер можна припустити, що для заданого набору N^1 несумісних рівнянь $Ax=h$ існує мінімальна зона захоплення Z , що має таку властивість, що будь-яка випадкова початкова послідовність $(x_0, x_1, \dots, x_k, \dots)$ майже напевно потрапляє в зону Z при $k \rightarrow \infty$. Звичайно, як тільки якийсь вектор x потрапить у зону Z , всі подальші вектори опиняться в цій зоні. Виразимо це припущення в строгій імовірнісній формі $p\{x_k \in Z\} = 1 - \varepsilon(k)$, де $\varepsilon(k) \rightarrow 0$ при $k \rightarrow \infty$.

У загальному випадку вірогідність існування послідовностей, які не потрапляють в зону захоплення при випадковому навчанні, дорівнює $\varepsilon(k) \leq Ac^k$, де A і $c < 1$ – константи, залежні від початкової точки x , і геометрії мінімальної зони захоплення.

Розмір і форма мінімальної зони захоплення виявляються дуже великими і складними навіть у простих випадках при невеликих значеннях p і N^1 , і тому може виявитися важливішим при теоретичних дослідженнях обмежитися гіперсферою, центр якої $x^* = A^+h$ визначається псевдозверненням за Муром – Пенроузом. У цьому випадку гіперсфера S мінімального радіусу R описує зону Z , і тоді можна сказати, що $p\{x_k \in Z\} = 1 - \varepsilon(k)$, де $\varepsilon(k) \rightarrow 0$ при $k \rightarrow \infty$. Мінімальна помилка, тобто найбільша довжина перпендикуляра x_k до N^1 гіперплощин, буде тоді менше $2R_0$.

На рис. 2.7 показано прості зони захоплення і відповідні мінімальні зони захоплення.

З погляду класичного алгоритму Качмажа зони захоплення відповідають областям збіжності за наявності перешкод вимірювань. Так, за наявності перешкод вимірювань $\xi(k) \sim N(0, \sigma_\xi^2)$ у вихідному сигналі $y(k+1)$, тобто

$$\tilde{y}(k+1) = y(k+1) + \xi(k+1), \quad (2.23)$$

алгоритм Качмажа збігається в область, визначену формулою

$$\lim_{k \rightarrow \infty} M \left\{ \left\| w(k+1) - w^* \right\|^2 \right\} = \frac{N}{N-2} \frac{\sigma_\xi^2}{\sigma_x^2}, \quad (2.24)$$

де $k \rightarrow \infty$, $M\{\bullet\}$ – символ математичного очікування; N – розмірність вектора вхідного сигналу $x \sim N(0, \sigma_x^2)$.

При цьому $\lim_{k \rightarrow \infty} M \left\{ w(k+1) - w^* \right\} = 0$, тобто оцінка вектора ваг є незміщеною.

Зменшити розмір області можна, модифікуючи алгоритм Качмажа так:

$$w(k+1) = w(k) + \gamma \frac{y(k+1) - w^T(k)x(k+1)}{\|x(k+1)\|^2} x(k+1), \quad (2.25)$$

де $\gamma \in (0, 2)$ – параметр, що впливає на швидкість навчання.

Введення параметра γ у алгоритм змінює траєкторію оцінки. Відповідні траєкторії за відсутності перешкод вимірювань показано на рис.2.2. Тут рис.2.1,б відображує випадок $0 < \gamma < 1$, а рис.2.1,в – випадок $1 < \gamma < 2$.

За наявності ж перешкод вимірювань така модифікація дозволяє забезпечити область збіжності, розмір якої визначається виразом

$$\lim_{k \rightarrow \infty} M \left\{ \|w(k+1) - w^*\|^2 \right\} = \frac{\gamma}{(2-\gamma)} \frac{N}{(N-2)} \frac{\sigma_\xi^2}{\sigma_x^2}. \quad (2.26)$$

Звідси видно, що приймаючи $\gamma \ll 1$, можна зменшити розмір області збіжності, а вибираючи параметр γ змінним, тобто задовольняючим умови Дворецького

$$\lim_{k \rightarrow \infty} \gamma(k) = 0; \sum_{k=1}^{\infty} \gamma(k) = \infty; \sum_{k=1}^{\infty} \gamma^2(k) < \infty, \quad (2.27)$$

можна отримати збіжність в точку, тобто

$$\lim_{k \rightarrow \infty} M \left\{ \|w(k+1) - w^*\|^2 \right\} = 0.$$

Модифікований алгоритм Альбуса, відповідний (2.25), матиме вигляд

$$w(k+1) = \begin{cases} w(k) + \gamma \frac{y(k+1) - w^T(k)x(k+1)}{\|x(k+1)\|^2} x(k+1); \\ w(k), \forall i, a_i(k) = 0. \end{cases} \quad (2.28)$$

Методи скорочення процесу навчання мережі. Однією з найважливіших характеристик алгоритму Качмажа є його швидкість збіжності, що визначає кількість кроків навчання (кількість ітерацій), необхідних для знаходження розв'язань системи рівнянь (2.15) із заданою точністю. Збіжність алгоритму істотно залежить від властивостей вхідної послідовності $a(k)$. Так, зокрема, якщо послідовність $a(k)$ є послідовністю взаємно ортогональних векторів, то розв'язання w^* можна знайти за кінцеве p число ітерацій, де p – розмірність вектора w . В інших випадках швидкість збіжності алгоритму Качмажа може виявитися недостатньою для знаходження і відстеження закону зміни розв'язання системи рівнянь (2.15) $w^*(k)$ у нестационарному випадку. У зв'язку з цим виникає необхідність в підвищенні швидкості збіжності алгоритму.

Швидкість збіжності алгоритму Альбуса визначається найбільшим за модулем власним значенням матриці $B = -(I_N + L)^{-1}L^T$ порядку N . При великих N ці власні значення обчислити важко. Функція Ляпунова дозволяє знайти оцінку швидкості збіжності у вигляді $[1 - 1/\lambda_{\max}(P)]^{1/2}$, де $\lambda_{\max}(P)$ позначає найбільше (дійсне, позитивне) власне значення матриць $P = (I_N + L^T)(I_N + L + L^T)^{-1}(I_N + L)$, яке також важко обчислити при великих N . Єдиною перевагою тут є те, що матриця P симетрична, тоді як матриця B такою не є. Результати чисельного моделювання і знаходження власних значень при невеликих N дозволяють припустити, що у багатьох випадках власні значення матриці B близькі $k+1$, залишаючись усередині одиничного кола, і тому швидкість збіжності цього алгоритму виявляється невеликою.

У зв'язку з цим виникає завдання прискорення процесу навчання мережі. При розгляді цього питання необхідно відзначити таке.

У загальному випадку вхідним сигналом для СМАС є вектор $x \in R^n$, що кодується в нормалізований вектор асоціацій \mathbf{a} . Проте для подальшого вивчення істотним є визначення вхідного сигналу як скаляр x , тобто $x \in R^1$. Тоді код x буде записаний так: $\hat{\mathbf{a}} = \hat{\mathbf{a}}(w) = (\hat{a}_1, \hat{a}_2, \dots, \hat{a}_p)^T$, де елементи $\hat{a}_i(w)$ отримані з

$$\hat{a}_i = \begin{cases} \frac{1}{\sqrt{\rho}}, & \text{якщо } w+1 \leq i \leq w+p, \\ 0, & \text{інакше.} \end{cases}$$

У прийнятому в даній роботі підході, тестування кожного алгоритму навчання починається з того, що мережу навчають відновлювати функцію одного змінного, тобто $n = 1$, незважаючи на те, що даний тест є достатньо простим в порівнянні з реальними системами з великими значеннями n , N і p . Такі тести проводилися й іншими авторами. Ті алгоритми, що пройшли цей тест, потім тестуються на n -вимірних вхідних сигналах x з $n = 2, 3, 4$.

Вихідний сигнал мережі на вхідний сигнал x , що поступив, після k кроків навчання обчислюється як середнє значення активних ваг, які визначаються ненульовими елементами вектора $\hat{\mathbf{a}}(s)$, за допомогою такого виразу:

$$y(x, k) = \frac{1}{\sqrt{\rho}} \hat{\mathbf{a}}_i^T(s) * w_k.$$

Навчання за допомогою ковзаючого середнього (AV). У даному алгоритмі значення вектора ваг $w(k)$ обчислюється як середнє значення корекцій, які отримано в базовому алгоритмі AL.

Для обчислення середнього використовується вектор-лічильник $c = (c_1, c_2, \dots, c_p)^T$ такої ж розмірності, як і вектор ваг $w(k)$. Початкове значення $c_0 = 0$.

Значення лічильника оновлюється в кінці k -го кроку навчання для того, щоб показати, як часто кожна вага була модифікована, так:

$$c(k+1) = c(k) + \sqrt{\rho} \hat{a}(k+1),$$

де $\sqrt{\rho} \hat{a}(k+1)$ містить одиницю у позиції активної ваги і нуль – інакше.

Настройка ваг здійснюється так:

$$w(k) = w(k-1) + \mu b(k), \quad (2.29)$$

$$\text{де } \mu(k) = \sqrt{\rho} r(k) - \hat{a}^T(k) w(k-1), \quad b(k) = (b_1(k), \dots, b_p(k))^T, \quad b_i(k) = \frac{\hat{a}_i(k)}{c_i(k)},$$

$\hat{a}^T(k)$ – скорочений запис $\hat{a}^T(x(k))$, а $\hat{a}_i^T(k)$ – i -й елемент даного вектора.

Відзначимо, що: 1) випадковий шум з нульовим математичним очікуванням, який часто зашумляє навчальні дані y_j і присутній в реальних системах, – відфільтрований; 2) якщо система рівнянь (2.15) несумісна, алгоритм збігається в точку, а не до обмеженого циклу, як в алгоритмі AL. Проте формального доказу збіжності даного алгоритму на сьогодні немає.

Алгоритм AV (2.25) збігається швидше, ніж AL при циклічному навчанні, але надалі збіжність стає дуже повільною для незашумлених тестів. У разі зашумленого навчального сигналу результати, отримані за допомогою

алгоритму AV, найкращі. Обчислювальні витрати невеликі і практично такі ж, як для алгоритму AL.

Навчання в точці з максимальною помилкою (ME). Цей метод застосовується разом з нормальною процедурою навчання в тій точці вхідного простору $x(j)$, для якої бажаний вихідний сигнал $y^*(j)$, отриманий з найбільшою помилкою серед заданої кількості l , останніх навчальних пар. Шукана довжина l вибирається в інтервалі $1 \leq l \leq k$. Минулі і справжні навчальні пари розміщуються в тимчасовій пам'яті. Відповідна помилка в точці $x(j)$ обчислюється для $j = \overline{k-l+1, k}$ так:

$$\begin{aligned} e_j(k+1) &= f(x_{(j)}(k)) - y_{(j)}^* = \\ &= \frac{1}{\sqrt{\rho}} \hat{a}_i^T(s_{(j)}) x(k) - y_{(j)}^* \end{aligned}$$

або, використовуючи скорочений запис,

$$y_j(k) = \frac{1}{\sqrt{\rho}} \hat{a}_{(j)}^T x_{k-1} - y_{(j)}^*.$$

Індекс $(j) = m$ найбільшої помилки визначається так, що

$$|e_{mk}| \geq |e_{(j)k}| \text{ для } j = k-l+1, \dots, k.$$

Якщо існує декілька найбільших помилок, тобто

$$|e_{m_1 k}| = |e_{m_2 k}| = \dots = |e_{m_\mu k}| \geq |e_{jk}|,$$

то вибирається найближча, тобто помилка m_j , відповідна найбільшому значенню j .

Корекція ваг для точки x_m відбувається так:

$$w(k+1) = w(k) + \mu(k)\hat{a}_m, \quad (2.30)$$

$$\mu_k = \sqrt{\rho}r_m - \hat{a}_m^T w_{k-1}.$$

Відзначимо, що при $l = 1$ цей алгоритм також ідентичний AL. Подальше вивчення швидкості збіжності даного алгоритму порівняно з AL, циклічним і випадковим навчанням, було проведено Еллісоном, який також рекомендує даний ME алгоритм.

Алгоритм ME (2.30) показує гарні властивості збіжності практично для всіх тестів. Початкове значення коефіцієнта навчання достатньо велике, іноді навіть більш, ніж для алгоритму GS (див. нижче). Для випадку відсутності перешкод, кращі властивості збіжності виходять при більшому значенні l . Проте велике значення l не завжди підходить для зашумлених даних. Обчислювальні витрати помірні і зростають лінійно з l .

Часткова оптимізація довжини кроку в останньому перпендикулярному напрямі (OS). У даному алгоритмі, як і в алгоритмі AL, використовується корекція вектора ваг у напрямі перпендикуляра до $\hat{a}(k)$, проте довжина кроку $\mu(k)$ вибирається не такою, що відразу забезпечує досягнення k -ї гіперплощини $\hat{a}^T(k)x = \sqrt{\rho}r_{(k)}$, а такою, що мінімізує суму квадратів перпендикулярів до гіперплощини, визначеної останніми l частинами навчальних даних, тобто критерій, що мінімізує

$$I_t = \sum_{i=k-j+1}^k (\hat{a}_{(j)}^T x_k - \sqrt{\rho}r_{(j)})^2 \rightarrow \min,$$

де l – ціле, що лежить в інтервалі $1 \leq l \leq k$.

Перпендикулярна відстань пропорційна відповідній помилці СМАС.

Останні l частин навчальних даних $(x(j), y^*(j)), j = \overline{k-l+1, k}$ (або в еквівалентному записі $(w(j), y^*(j))$) записуються в тимчасову пам'ять.

Ваги настроюються так:

$$w(k+1) = w(k) + \mu(k+1)u(k+1),$$

$$\mu(k+1) = \frac{\sum_{i=k-j+1}^k (\sqrt{\rho}r(k+1) - \hat{a}_{(j)}^T x(k)) \hat{a}^T(j) \hat{a}(k)}{\sum_{i=k-j+1}^k (\hat{a}^T(j) \hat{a}(k))^2}. \quad (2.31)$$

Відзначимо, що базовий алгоритм AL – окремий випадок даної процедури з $l = 2$. Для цього алгоритму також немає формального доказу його збіжності.

Алгоритм OS (2.31) добре підходить для зашумлених гладких тестових функцій. За його допомогою можна отримати в цьому випадку високу точність при великому значенні l . Для незашумленого випадку швидкість збіжності відносно повільна і зменшуватиметься зі збільшенням l . Обчислювальні витрати більші, ніж для алгоритму ME (2.30).

Цікаві результати можна отримати при комбінації різних алгоритмів, наприклад, ME+AV. Загальну збіжність можна прискорити обмеженням значення лічильника, наприклад, $c_i \leq c_{\max}$. Ця ідея досліджуватиметься в подальших роботах.

Сьогодні рекомендується використовувати алгоритм ME (2.27), оскільки він досягає хорошої збіжності для широкого спектру тестів з помірними обчислювальними витратами.

Алгоритм навчання з самоналагоджувальними коефіцієнтами навчання. Відомо, що покращений алгоритм з самоналагоджувальним коефіцієнтом навчання, який реалізується так:

а) знайти межі збіжності і отримати коефіцієнт навчання, що призводить до максимальної швидкості збіжності β_{\max} :

– встановити максимальну кількість епох пошуку \max_epoch1 і початкове значення коефіцієнта навчання β_0 , потім $\beta \leftarrow \beta_0$ і обчислити початкову вихідну помилку pre_Mse2 . Встановити $epoch = 1$ і $flag = 0$;

– оцінити поточну вихідну помилку СМАС $Mse1$, потім хай $\beta_0 \leftarrow \beta$;

– якщо $Mse1 \geq pre_Mse1$, то продовжити, інакше перехід до 5);

– якщо $flag = 0$, хай $\beta \leftarrow \frac{\beta + \beta_0}{2}$; якщо $flag = 1$, хай $flag \leftarrow 1$. Перехід до

б);

– якщо $flag = 0$, хай $\beta \leftarrow \frac{\beta + \beta_0}{2}$, $flag \leftarrow 1$, і перехід до б); якщо $flag = 1$, хай $\beta_{\max} \leftarrow \beta$, вихід $-\beta_{\max}$, завершити крок а) і перехід на крок б);

– $epoch \leftarrow epoch + 1$. Якщо $epoch \leq \max_epoch1$, перехід на 2); інакше закінчити програму, видати повідомлення про помилку, збільшити кількість епох пошуку;

б) пошук оптимального значення коефіцієнта навчання β_{best} в інтервалі $(0, \beta_{\max})$.

Послідовність $\{x_i\}, i = 0, 1, 2, \dots$, що згенерована за допомогою Logistic mapping,

$$x_{n+1} = 4x_n(1 - x_n), n = 0, 1, 2, \dots \quad (2.32)$$

буде випадково розподілена в інтервалі $[0, 1]$ при $0 \leq x_0 \leq 1$. Встановити h , і використовуючи вираз (4.20), згенерувати послідовність $\{x_i\}, i = 0, 1, 2, \dots, h-1$ з

початковим значенням $x_0 = 0.1$, і виконати перетворення $x_i = \beta_{\max} x_i$, потім, послідовність після перетворення буде випадково розподілена в інтервалі збіжності $(0, \beta_{\max})$. Встановити параметр max_epoch2 , відповідно використовуючи $x_i, i = 0, 1, 2, \dots, h-1$ як коефіцієнт навчання для пошуку max_epoch2 епох. Порівнюючи з вихідними помилками відповідними x_i , вибрати x_i , яке робить помилку мінімальною і виконати $\beta_{\text{best}} \leftarrow x_i$, і потім перейти до кроку в);

в) використовуючи β_{best} як коефіцієнт навчання, продовжити традиційне навчання СМАС впродовж max_epoch2 епох. Закінчити програму, досягши бажаного рівня помилки або після закінчення максимальної кількості епох.

Рекомендовані значення параметрів для запропонованого алгоритму такі: $\beta_0 = 2$; $\text{max_epoch1} = 100$; $h = 20$; и $\text{max_epoch2} = 100$.

2.1.4 Застосування нейронної мережі СМАС для обробки відеоінформації

Будь-яке плоске зображення за своєю суттю є деякою функцією колориметричної характеристики точки від її координат. Під колориметричною характеристикою можна розуміти яскравість і кольоровість точки в системі YUV, яскравість колірних складових у системі RGB, яскравість для зображень у відтінках сірого кольору. Математично задати будь-яке зображення реальних об'єктів практично неможливо внаслідок його складності. Класичні методи апроксимації статечними поліномами не дадуть бажаної якості, крім того, обчислення коефіцієнтів цих поліномів – у край складне і ресурсомістке завдання. Табличне подання зображень забезпечить максимальна якість, але водночас зажадає великих витрат пам'яті і пропускну здатності каналів зв'язку для зберігання і передачі. Вихід був знайдений в стисненні зображень методами архівації без втрат і з втратами. Проте такі методи або добре працюють тільки на вузьких класах зображень, або вимагають великих обчислювальних витрат для компресії і декомпресії. Ці методи засновані на особливості зображень,

пов'язаній із залежністю сусідніх елементів. Подібна властивість безлічі вхідних сигналів використовується під час обчислення функцій розглянутою нейронною мережею СМАС, що дозволяє запропонувати її як засіб для стиснення і фільтрації зображень. Крім того, особливості структури мережі СМАС, такі як вибірка обмеженої кількості ваг з пам'яті на даному кроці, істотне зниження об'єму пам'яті для зберігання параметрів, роблять привабливим використання мережі СМАС при апаратній реалізації обробки зображень.

Далі розглянемо застосування мережі СМАС для реалізації обробки зображень. При цьому необхідно врахувати, що стандартне телевізійне зображення має розміри 720x576 точок, тобто кількість рівнів квантування для вхідних змінних, відповідних координатам точки (x,y) в зображенні, значення інтенсивності, в якій обчислює мережу, неоднакове. В цьому випадку залежність (2.3) для розрахунку об'єму пам'яті, необхідного для кодування зображення мережею СМАС, набуде вигляду:

$$n_{\max} = \left\lceil \rho \left(\frac{R_x - 1}{\rho} + 1 \right) \left(\frac{R_y - 1}{\rho} + 1 \right) \right\rceil, \quad (2.33)$$

де R_x і R_y – кількість рівнів квантування для змінних R_x і R_y відповідно. Кількість ступенів квантування вважатимемо однаковою для обох змінних.

Застосування мережі СМАС при мікшуванні зображень. Операцію мікшування зображень в колориметричній системі RGB можна описати виразом:

$$\begin{cases} R_O(x, y) = K_{R1}(x, y)R_{I1}(x, y) + K_{R2}(x, y)R_{I2}(x, y) \\ G_O(x, y) = K_{G1}(x, y)G_{I1}(x, y) + K_{G2}(x, y)G_{I2}(x, y), \\ B_O(x, y) = K_{B1}(x, y)B_{I1}(x, y) + K_{B2}(x, y)B_{I2}(x, y) \end{cases}, \quad (2.34)$$

де буквенний індекс визначає приналежність до колірної каналу, а цифровий індекс – до шини даних.

Для системи YUV замість інтенсивностей колірних складових необхідно підставити значення яскравості і кольорорізницевих сигналів. Ця операція лежить в основі переважної кількості переходів між сюжетами у відеопотоці. Форма переходу, фігура або шторка, задається функцією присутності $K(x,y)$ і від можливостей системи обробки за реалізацією цієї функції залежить різноманітність можливих переходів. Табличне задання цієї функції вимагає витрат часу для зміни шторки або фігури, апаратне формування функції присутності обмежує коло можливостей відеопроцесора. Одним з можливих шляхів скорочення об'єму пам'яті, потрібної для задання функції присутності, є застосування для її апроксимації нейронної мережі СМАС.

За своєю суттю функція присутності – це деяке зображення, яке назвемо «фігура мікшування», за розмірами таке ж, як мікшоване зображення, яскравість точок якого визначає ступінь присутності початкових зображень в результуючому. Так, чим яскравіше точка фігури, тим більше в результуючому зображенні в цій же точці присутнє перше зображення і відсутнє друге. На рис.2.6 наведено мікшоване зображення а) і б), фігура мікшування в), задана у відтінках сірого, і результат мікшування г) початкових зображень з використанням фігури. Геометрично функція присутності для задання зображеної на рис.2.6 фігури є конічною поверхнею, яку аналітично можна подати у вигляді:

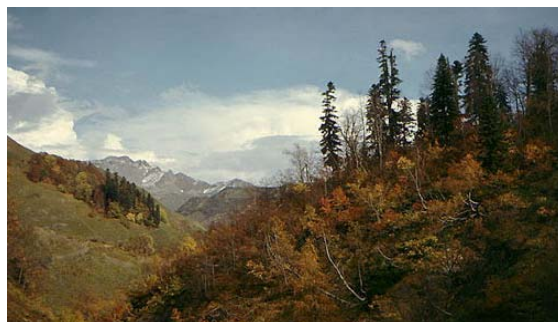
$$\begin{cases} \frac{(K(x,y) - K_0)^2}{c^2} = \frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2}, \\ K(x,y) = 0, K(x,y) = 255 \end{cases}, \quad (2.35)$$

де параметри a, b, c визначають еліптичність і гостроту конічної поверхні, x_0, y_0, K_0 – положення осі і вершини конічної поверхні.

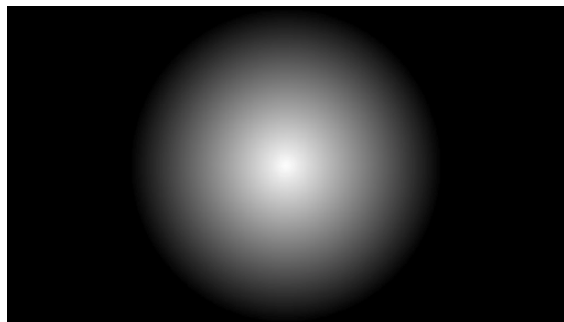
Площини $K(x,y) = 0$ та $K(x,y) = 255$ обмежують область припустимих значень функції присутності. Тут і далі вважатимемо, що повній присутності зображення 1 і відсутності зображення 2 відповідає значення функції присутності 255, а повній відсутності зображення 1 і присутності зображення 2 – значення 0.



а



б



в



г

Рисунок 2.6 – Приклад мікшування зображень

Перевагою використання мережі СМАС для апроксимації функцій присутності є гнучкість, властива табличному способу задання функції і при цьому істотно менший необхідний об'єм пам'яті. Якщо зображення фігури має розмір 720×576 точок, об'єм пам'яті, необхідний для табличного її задання складе $720 \cdot 576 = 414720$ байт. Під час використання мережі СМАС з кількістю ступенів $\rho = 4$ цей об'єм складе 104655 байт, а при $\rho = 8$ – 52981 байт, що, відповідно, в 3,96 і в 7,83 разів менше, порівняно з табличним способом. Наведені значення можуть бути зменшені під час використання хешування в мережі СМАС.

Під час використання мережі СМАС для синтезу функцій присутності в реальному для відеопотоку часі не потрібне навчання мережі безпосередньо в процесі використання. Його можна провести на математичній моделі мережі,

реалізованої у вигляді програми на управляючій ЕОМ. Під час навчання як бажані значення функції присутності мережі можуть подаватися як обчислені за математичними залежностями значення функції присутності, так і отримані шляхом цифрування елементи розроблених дизайнером фігур. Отримані в результаті навчання вагові коефіцієнти передаються інтерфейсом з ЕОМ у відеопроцесор, де зберігаються в пам'яті, адреси читання якої формуються при апаратному кодуванні координат.

Виконання колірної корекції зображень з використанням мережі СМАС. Вище розглянуто спосіб колірної корекції зображень для випадку залежності передавальних функцій коректора тільки від інтенсивностей колірної складової в даному каналі. Головний недолік такого способу корекції в тому, що він не дозволяє усувати такі спотворення, як появи небажаних відтінків на ділянках зображень певного кольору. Для такої корекції необхідно мати передавальні функції, залежні від всіх колірних складових:

$$\begin{aligned} R_{out} &= f_R(R_{in}, G_{in}, B_{in}) \\ G_{out} &= f_G(R_{in}, G_{in}, B_{in}), \\ B_{out} &= f_B(R_{in}, G_{in}, B_{in}) \end{aligned} \quad (2.36)$$

де R_{in} , G_{in} , B_{in} – вхідні, а R_{out} , G_{out} , B_{out} – вихідні значення інтенсивностей кольорів коректора.

При цьому табличний метод вимагає $2^{3 \cdot 8} = 16777216$ байт пам'яті на один колірний канал при поданні інтенсивностей кольорів 8-бітовими кодами. При синтезі передавальних функцій мережею СМАС необхідно 1085874 і 284242 байти пам'яті при $\rho = 4$ і $\rho = 8$. Економія складе 15,45 і 59 разів відповідно.

Запропоновано використовувати нейронну мережу СМАС для колірної корекції зображень, що поступають з сканера на принтер. Розглянуто блокові структури для такої корекції. Проте детальну роботу блоків задання функцій корекції не розкрито. Крім того, реальний час для розглянутого завдання

складає декілька десятків секунд на кадр (час проходження сканера). Внаслідок цього, модель колірної корекції зображень, що використовує нейронну мережу СМАС для апроксимації передавальних характеристик, потребує подальшого розвитку, пов'язаного з її адаптацією для роботи в реальному для відеопотоку часі.

Безпосередньо у колірному коректорі необхідно реалізувати мережу СМАС, що має три вхідні змінні і пам'ять ваг, що заповнюється значеннями, обчисленими при навчанні моделі мережі на управляючій ЕОМ. Задання навчальних дій у формі, зрозумілій для користувача відеопроцесора, є складним завданням, оскільки візуально необхідно подати функцію від трьох змінних. Можна запропонувати такий спосіб, який розглянемо на прикладі червоного колірного каналу:

1. Будується передавальна характеристика $f_R(R_{in}, 0, 0)$ шляхом графічного задання додаткових точок на графіці в системі координат (R_{in}, f_R) так, як наведено на рис.2.7 (на рис.2.7 осі позначено R_i і R_o відповідно);

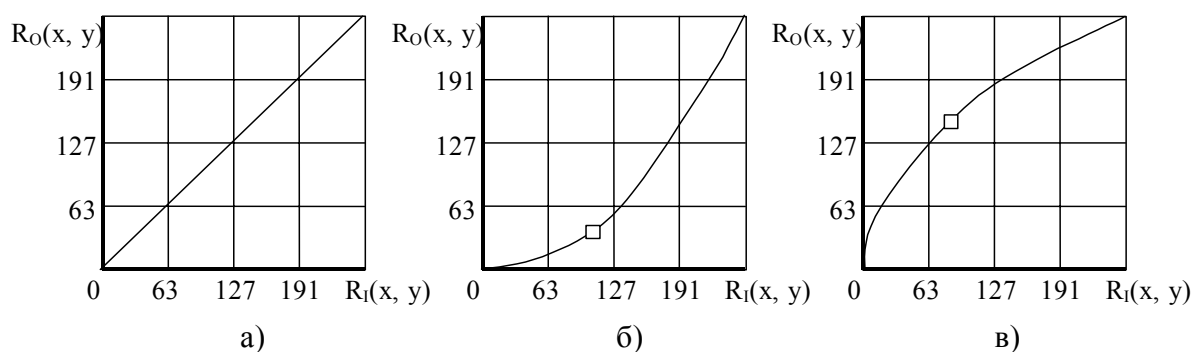


Рисунок 2.7 – Задання передавальних характеристик колірної корекції

2. У системі координат (G_{in}, B_{in}) задаємо положення точок, в яких інтенсивність червоного кольору має залежати від значень інтенсивності зеленого і синього кольорів.

3. У системах координат (G_{in}, dR) (B_{in}, dR) задаються величини і знак залежності червоного від зеленого і синього кольорів відповідно.

4. Отримуємо декілька точок передавальної характеристики для каналу червоного кольору:

$$R_{out} = f_R(R_{in}, 0, 0) + dR(G_{in}) + dR(B_{in}). \quad (2.37)$$

2. Подаємо отримані точки як навчальні програмної моделі мережі СМАС.

Зважаючи на згладжуючі властивості мережі СМАС, даний метод задання передавальних характеристик колірною коректора має забезпечити плавні передавальні характеристики, що не мають розривів і різких стрибків.

Застосування мережі СМАС для стиснення і фільтрації зображень. Подібно до того, як функцію присутності $K(x,y)$ можна подати у вигляді зображення, що має в точці (x,y) значення яскравості, рівне значенню функції присутності, будь-яке зображення можна подати у вигляді функцій, що визначають значення його колірних компонентів в кожній точці:

$$\begin{cases} R = f_R(x, y) \\ G = f_G(x, y) \\ B = f_B(x, y) \end{cases} \hat{=} \begin{cases} Y = f_Y(x, y) \\ U = f_U(x, y) \\ V = f_V(x, y) \end{cases}. \quad (2.38)$$

Вид функцій $f(x,y)$ визначається змістом зображення і математично, в переважній більшості випадків, не подається. Табличне подання зображень вимагає великих об'ємів пам'яті і швидкостей вводу-виводу та передачі для зберігання і передачі відеоінформації, хоч і забезпечує максимальну її якість. Всі існуючі алгоритми стиснення припускають, що елементи зображень залежні між собою, а сусідні кадри у відеопотоці відрізняються положенням деяких об'єктів. Багато хто з них видаляє, вважаючи надмірними, високі частоти, присутні в зображенні, хоча ці складові впливають на різкість і чіткість меж об'єктів. Алгоритми, що мають високий ступінь стиснення, або істотно погіршують якість зображень, або вимагають великих обчислювальних витрат.

Якщо розглядати зображення як деяку функцію від координат, задачу стиснення можна розв'язати шляхом апроксимації цієї функції, способом, що вимагає менше пам'яті для зберігання коефіцієнтів. Таку властивість має дана нейронна мережа СМАС.

Метод стиснення зображення мережею СМАС полягає в такому. Вхідними сигналами мережі є координати (x,y) точок зображення, що стискається, на екрані. Ці значення кодуються вхідним шаром мережі, формуючим вектор асоціацій. Відповідно до цього вектора, з пам'яті зчитуються ρ ваг, які з урахуванням значень базисних функцій першого шару підсумовуються в третьому шарі. Набуте значення функції порівнюється із значенням інтенсивності в даному колірному каналі зображення. Отримана помилка використовується як критерій підстроювання ваг алгоритмом навчання, якому необхідно модифікувати ті ваги, які були використані під час обчислення мережею значення функції.

Для опису процесу стиснення зображення позначимо через $I_{in}(x,y)$ функцію, що виражає залежність від координат інтенсивності однієї з складових початкового зображення. В результаті апроксимації отримаємо зображення $I_1(x,y)$, пов'язане з початковим співвідношенням:

$$I_{in}(x,y) = I_1(x,y) + e_1(x,y), \quad (2.39)$$

де $e_1(x,y)$ – помилка, що отримується при апроксимації. Цю помилку можна подати як деяке різницеве зображення, що отримується шляхом віднімання початкового і апроксимованого зображень. Це зображення, подібно до початкового, може апроксимуватися мережею СМАС, внаслідок чого отримуємо зображення $I_2(x,y)$ і помилку $e_2(x,y)$:

$$e_1(x,y) = I_2(x,y) + e_2(x,y). \quad (2.40)$$

Під час використання n мереж СМАС, і, відповідно, n кроків апроксимації, маємо співвідношення, що отримується з підстановки (2.40) в (2.39) і узагальнення на n кроків:

$$I_{in}(x, y) = \sum_{j=0}^n I_j(x, y) + e_n(x, y). \quad (2.41)$$

Особливістю (2.41) є те, що воно визначає перетворення початкового зображення без втрат. Якщо на n -му кроці апроксимації помилкою нехтувати, отримаємо співвідношення

$$I_{in}(x, y) = \sum_{j=0}^n I_j(x, y). \quad (2.42)$$

Кількість кроків n і кількість ступенів квантування на кожному кроці апроксимації визначають якість і ступінь стиснення зображення.

Особливістю пропонованого методу стиснення зображень є те, що якість апроксимації залежатиме від кількості поданих під час навчання мережі точок зображення. Ці точки можуть вибиратися в початковому зображенні як за будь-яким критерієм (межею, областю, об'єктом), так і випадково за допомогою будь-якого з алгоритмів формування псевдовипадкової або випадкової послідовності. Максимальна якість буде досягнута після подання мережі всіх точок зображення.

Слід відзначити, що під час обробки зображення розгортається в часі рядками і стовпчиками. Проте таке систематизоване подання навчальних точок призводить до появи направлених «змазувань» у відновленому зображенні. Для усунення цього недоліку необхідно мати проміжний буфер, який записується послідовно, а читається випадково.

Структуру кодера-декодера зображень, заснованого на застосуванні апроксимації функцій мережею СМАС, наведено на рис. 2.8.

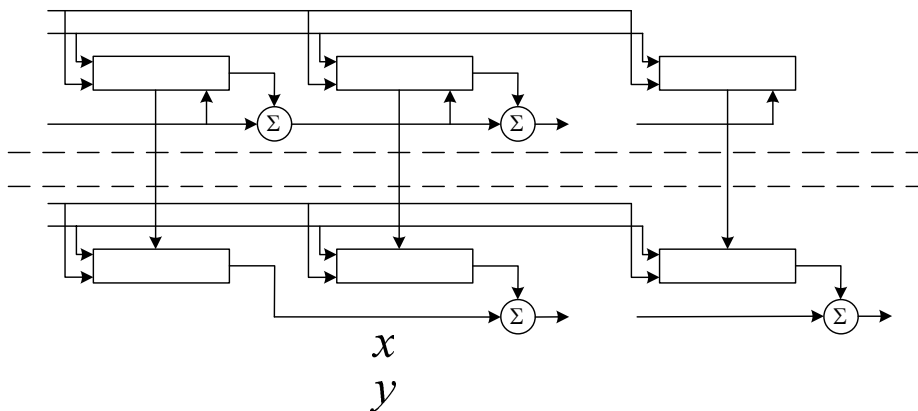


Рисунок 2.8 – Ієрархічна структура СМАС для кодування-декодування зображень

На наведеному рисунку мережі СМАС є вищерозглянуті структури, що складаються з вхідного кодуєчого шару, асоціативного шару і вихідного обчислюючого шару. Відмінність мереж СМАС кодера полягає в тому, що вони мають засоби для настройки вагових коефіцієнтів. Ці коефіцієнти і є закодоване (стисле) зображення, яке можна зберігати, або передавати лінією зв'язку. У декодері ці вагові коефіцієнти заносяться в асоціативну пам'ять відповідних мереж і беруть участь в обчисленні значень функцій. Отже, декодуюча сторона простіша і дешевша в реалізації, що є одним з важливих атрибутів алгоритму стиснення у зв'язку з тим, що декодування виконується в апаратурі користувачів відеоінформації, яких, як правило, в декілька тисяч разів більше, ніж об'єктів, що формують відеосигнал.

Іншою важливою особливістю використання мережі СМАС для апроксимації зображень є можливість фільтрації зображень. Розглянемо залежність (2.39). Якщо припустити, що в цій залежності зображення $I_1(x,y)$ містить частина частот початкового зображення, то в цьому випадку помилка $e_1(x,y)$ міститиме ті частоти, які присутні в початковому зображенні і відсутні в обчисленому зображенні $I_1(x,y)$. В цьому випадку мережа СМАС буде фільтром тих або інших частот, а отже, може використовуватися для згладжування і

виділення контурів, очищення від шуму, виділення деталей, що з'явилися. Частоти, що виділяються мережею, визначаються параметром ρ і точками початкового зображення, поданими мережі для навчання. Чим вище ρ – тим більше низькочастотні складові виділить мережа, при цьому для навчання мережі необхідно подати в ρ^2 менше точок, ніж в зображенні. Для фільтрації може застосовуватися структура, наведена на рис. 2.8. Ця структура дозволить виділяти ті або інші групи частот (кожна мережа відповідає за певну групу частот, відповідно до якої вибираються її параметри і проводиться навчання).

2.1.5. Модифіковані архітектури ШНМ СМАС

ШНМ СМАС була запропонована Альбусом ще в 1975 р. У цей час мережі СМАС успішно застосовуються в багатьох сферах життєдіяльності суспільства, таких як роботизоване керування, обробка сигналів, збір даних у глобальній комп'ютерній мережі, а також розпізнавання образів. Поширенню нейронних мереж СМАС чимало сприяла здатність до швидкого навчання, гарна здатність до узагальнення й простота апаратної реалізації.

Попередній досвід вивчення ШНМ СМАС полягав у спробах створення розширених алгоритмів навчання СМАС, оптимізації структури мережі СМАС, виборі параметрів навчання для підвищення швидкості збіжності, апаратної реалізації моделі СМАС, а також застосуванні нейронних мереж СМАС у різних сферах. Властивості збіжності СМАС доведені математично.

Для навчання ШНМ СМАС застосовується локальний навчальний підхід, а саме: кожна точка навчальних даних асоціюється з кінцевим набором ваг, спільно використовуваних сусідніми точками. Отже, між вагами, настроєними по одній точці навчальних даних, і сусідніми точками можуть виникати колізії.

В СМАС Альбуса використовуються константні (одичні) БФ для виконання локально-локально-зваженої апроксимації сигналів. У результаті застосування константних БФ настроювання вагових коефіцієнтів мережі здійснюється за допомогою рівномірного розподілу похибки на асоційованому

наборі ваг. У межах кожного дискретного стану виходом мережі СМАС є константа, у той час як інформація про похідні не зберігається.

Запропонованій Альбусом моделі СМАС властиві два основних недоліки: по-перше, для моделювання функцій багатьох змінних або при класифікації багатомірних образів потрібно мати надмірну кількість пам'яті; по-друге, існує проблема рівномірності дискретизації вхідного простору навчальних наборів.

Як показано вище, при моделюванні багатомірних функцій з ростом розмірності простору вхідних змінних обсяг пам'яті, необхідний для зберігання вагових коефіцієнтів класичної СМАС, зростає експоненційно. Крім того, збільшується кількість необхідних точок навчальних даних і зростає складність кодування інформації. Обмежені ж об'єми й швидкодія пам'яті в прикладних системах значною мірою звужують галузь практичного застосування СМАС у реальних застосуваннях.

У свою чергу, при реалізації рівномірного квантування вхідного простору навчальних сигналів доводиться ігнорувати проблему нерівномірності розподілу навчальних образів, у результаті чого виділяються невикористовувані згодом області пам'яті й знижується продуктивність навчання як наслідок неоптимального використання пам'яті для зберігання отриманих у процесі навчання даних.

Серед відомих способів вирішення описаної проблеми найбільш відомим є гешування, що полягає у відображенні більшої необхідної пам'яті в меншу, фізично реалізовану. Застосування геш-кодування істотно знижує вимоги ШНМ СМАС до пам'яті при рішенні деяких задач. Але при цьому неминуче виникає проблема геш-колізій, що полягає в появі небажаної кореляції між вхідними сигналами, що досить негативно позначається на апроксимуючих властивостях мережі, що може призвести до зниження швидкості збіжності й навіть розбіжності алгоритму навчання.

Для зниження вимог до пам'яті при моделюванні багатомірних задач була запропонована модульна СМАС, для якої з метою оптимізації обчислень

Альбусом застосований метод тимчасової інверсії. В СМАС з тимчасовою інверсією для вирішення багатомірних задач використовуються підключені послідовно СМАС меншої розмірності. При цьому модулі СМАС вихідного шару породжують тимчасові точки навчальних даних для настроювання параметрів СМАС малої розмірності, розташованих у нижніх шарах.

Так, для рішення багатомірних завдань Лином і Чи була запропонована нейромережева структура, що складається з малих двовимірних мереж СМАС. На практиці така структура показала недієздатність у випадку неправильного вибору архітектури модулів мережі. Фактично, метод Ліна й Чи схильний до проблеми перестановки вхідних змінних при вирішенні деяких задач класифікації образів з високою деталізацією вимірів (таких, як набір даних Лунга з 56-ю вхідними ознаками).

Для моделювання багатомірних гладких функцій Хангом і Яном використана нейронна мережа MS_СМАС. Мережа такого типу поєднує множину одномірних ШНМ СМАС у деревовидну структуру, що навчається за допомогою методу тимчасової інверсії. Крім того, Хангом і Яном розроблено схему квадратичного сплайна для згладжування значень виходів одномірної мережі СМАС за допомогою трансформації ступінчастих ваг у безперервно-гладкі ваги. Однак як недолік нейронної мережі MS_СМАС необхідно вказати нездатність вирішувати задачі класифікації багатомірних образів, оскільки її деревовидна структура швидко росте зі збільшенням розмірності вхідних векторів.

Спроби вирішення задачі пошуку оптимальних схем дискретизації вхідного простору сигналів у ШНМ СМАС здійснюються протягом досить тривалого часу, але дотепер її вирішення не втрачає актуальності. Так, з метою адаптивного розподілу пам'яті пропонується застосування в мережах СМАС декількох шарів дискретизації, що мають різну роздільну здібність.

Також використовується підхід з ітеративним зворотним зв'язком на виході СМАС для настроювання вхідної функції дискретизації й, як наслідок,

рівномірного розподілу цільових вихідних сигналів системи. Не слід забувати, що для використання ітеративного зворотного зв'язку необхідна похідна цільової функції за часом для корекції функції дискретизації, а це вкрай ускладнює процес навчання. Крім зазначеного недоліку, розглянутий підхід також не може бути застосовано для вирішення задач класифікації образів. З метою усунення описаного недоліку Бергером був запропонований адаптивний бінарний метод розподілу вхідного простору на прийнятні діапазони дискретизації вхідного простору при мінімізації помилки навчання.

У свою чергу, досліджене застосування кластеризації для одержання адаптивного вирішення дискретизації вхідного простору. На закінчення можна відзначити, що зазначені підходи завжди передбачають компроміс між економією сховища даних і обчислювальною складністю розглянутої моделі.

2.1.6. Адаптивна дискретизація вхідних сигналів

Адаптивне настроювання схеми дискретизації вхідних сигналів у нейронній мережі СМАС можуть здійснюватися на основі обчислення й аналізу коефіцієнта подібності Грея, що використовується для визначення міри подібності послідовностей.

Якщо задано еталонну послідовність $x_0 = \{x_0(1), x_0(2), \dots, x_0(n)\}$, з якою порівнюють m послідовностей виду $x_i = \{x_i(1), x_i(2), \dots, x_i(n)\}$, $i = 1, 2, \dots, m$, то коефіцієнт подібності Грея між x_0 і x_i на k -ому кроці визначається як

$$r(x_0(k), x_i(k)) = \frac{\Delta_{\min} + \zeta \cdot \Delta_{\max}}{\Delta_{0i}(k) + \zeta \cdot \Delta_{\min}}, \quad (2.43)$$

де $\Delta_{0i}(k) = |x_0(k) - x_i(k)|$, $\zeta \in (0, 1]$ – коефіцієнт відмінності Δ_{\max} й Δ_{\min} ,
 $\Delta_{\max} = \max_i \max_j \Delta_{0i}(k)$, $\Delta_{\min} = \min_i \min_j \Delta_{0i}(k)$. Як витікає з (2.43),
 $0 < r(x_0(k), x_i(k)) \leq 1$.

Після визначення всіх коефіцієнтів подібності Грея можна обчислити їх зважене середнє, що називають ступенем подібності Грея

$$r(x_0, x_i) = \sum_{k=1}^n [w_k \cdot r(x_0(k), x_i(k))], \quad (2.44)$$

де w_k – ваговий фактор, $\sum_{k=1}^n w_k = 1$ (звичайно вибирають $w_k = 1/n$ для всіх k).

При використанні адаптивної дискретизації навчання СМАС починається із заданою рівномірною схемою дискретизації. Нехай вхідний простір на початку розділено на n частин і бажана функція \hat{y} відома. Позначимо реакцію мережі (вихідний сигнал) на вхідний сигнал x_k ($k = 1, 2, \dots, n$), що надійшов, як $y(k)$. Об'єднання n бажаних виходів дозволяє сформувати еталонну послідовність $\hat{Y} = \{\hat{y}(1), \hat{y}(2), \dots, \hat{y}(n)\}$.

Для аналізу подібності Грея між бажаними й реальними вихідними сигналами мережі, порівнювана послідовність $Y = \{y(1), y(2), \dots, y(n)\}$ генерується реальними виходами СМАС на кожному кроці навчання. Відповідно до (2.43), коефіцієнт подібності Грея для сигналу x_k для однієї порівнюваної послідовності може бути визначений як

$$r(y(k), \hat{y}(k)) = \frac{\Delta_{\min} + \zeta \cdot \Delta_{\max}}{\Delta_y(k) + \zeta \cdot \Delta_{\min}}, \quad (2.45)$$

де $\Delta_y(k) = |y(k) - \hat{y}(k)|$, $\Delta_{\max} = \max_k \Delta_y(k)$, $\Delta_{\min} = \min_k \Delta_y(k)$ і $\zeta \in (0, 1]$ – коефіцієнт відмінності.

Внаслідок того, що при рівномірній дискретизації не враховується швидкість зміни бажаної функції \hat{y} , результати навчання можуть бути далекими від оптимальних. Пропонований метод адаптивної дискретизації

базується на мінімізації вихідної помилки СМАС, що виникла в результаті дискретизації вхідних сигналів. Як ступінь зміни бажаної функції можливе використання, наприклад, її похідної, що можна отримати заздалегідь. Однак, недоліком такого підходу є те, що коли бажана функція змінюється, то значення похідної також змінюється. З іншого боку, при розподілі вхідного простору на основі аналізу вихідної помилки виникає проблема визначення адекватної стратегії дискретизації.

Метод адаптивної дискретизації, що використовує коефіцієнт подібності Грея, не тільки відображає співвідношення між еталонною й порівнюваною послідовністю в стані x_k , але й характеризує вихідну помилку – чим більше коефіцієнт Грея, тим менше значення вихідної помилки.

Основною перевагою використання коефіцієнта подібності Грея є те, що він легко обчислюється наприкінці кожної ітерації навчання мережі. Якщо результат незадовільний, то відбувається корекція кроку дискретизації з використанням коефіцієнта подібності Грея у відповідності зі значенням функції ухвалення рішення про дискретизацію $Q(r)$ як показано на рис. 2.9.

Якщо в деякій області коефіцієнти подібності Грея менше деякого фіксованого значення r_c (межа збільшення розділення), то крок дискретизації в цій області зменшується, якщо ж вони більше, ніж r_d (межа зменшення розділення) – то збільшується. Якщо коефіцієнти подібності Грея знаходяться усередині інтервалу $[r_c, r_d]$, відповідні інтервали дискретизації не змінюються.

Нехай $r(k)$ – коефіцієнт подібності Грея в стані x_k . Якщо існує кілька послідовних станів x_i, x_{i+1}, \dots, x_j , що задовольняють умові, що всі їхні коефіцієнти подібності менше r_c , тобто $0 < r(k) < r_c$ для $k = i, i+1, \dots, j$, то число дискретизації в області, яка поєднує ці стани, позначене як $R_{ij} = [x_i^L, x_j^R]$, повинно збільшуватися для усунення помилки навчання.

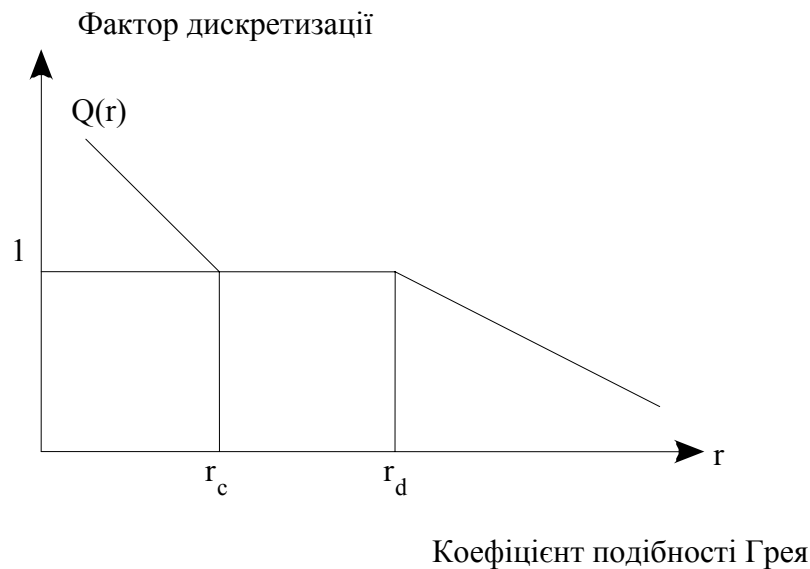


Рисунок 2.9 – Функція ухвалення рішення про дискретизацію

Нехай r_{ij} – середнє $r(k)$ для $k = i, i + 1, \dots, j$, тоді число дискретизації в об'єднаній області R обчислюється за допомогою функції ухвалення рішення про дискретизацію як

$$\Omega_{ij} =] Q(r_{ij}) \cdot N_x [, \quad (2.46)$$

де $] \bullet [$ позначає округлення до найближчого цілого; $Q(r_{ij}) > 1$ – значення функції ухвалення рішення про дискретизацію; $N_x = j - i + 1$ – представляє число послідовних станів, що задовольняють наведеним вище вимогам.

З виразу (2.46) видно, що $\Omega_{ij} > N_x$, тобто розділення в області R_{ij} збільшується. Також, крок дискретизації в області R_{ij} після адаптивного настроювання стає рівним

$$d_{ij} = \frac{x_j^R - x_i^L}{\Omega_{ij}}. \quad (2.47)$$

Внаслідок того, що дискретизаційне число в області R_{ij} змінюється з N_x на Ω_{ij} , число станів у вхідному просторі X стає рівним $n - N_x + \Omega_{ij}$, на відміну від вихідного n .

Інакше, якщо $r(k) > r_d$ для $k = i, i+1, \dots, j$, то вихідні сигнали мережі в цих станах близькі або дуже близькі до їх відповідних бажаних значень. У цьому випадку крок дискретизації в об'єднаній області R_{ij} повинен бути збільшений для зменшення розділення в цій області. Відповідно до функції ухвалення рішення про дискретизацію на рис. 2.9, можна помітити, що в розглянутому випадку $0 < Q(r_{ij}) < 1$, а тому дискретизаційне число Ω_{ij} , визначене в (2.46), повинно задовольняти $\Omega_{ij} < N_x$, що відповідає зменшенню розділення в області R_{ij} . У цьому випадку кількість станів у вхідному просторі X після адаптивного настроювання дискретизації стане рівним $n - N_x + \Omega_{ij}$, що менше вихідного n . Крок дискретизації обчислюється також за допомогою (2.47).

Слід відзначити, що наведені вище ствердження справедливі тільки до послідовностей станів, а не до одиничних станів. При розгляді одиничного стану можна припустити, що $i = j$. Якщо коефіцієнт подібності Грея в цьому стані менше r_c , то метод адаптивної дискретизації може бути застосований для збільшення розділення дискретизації. Але збільшити інтервал дискретизації для зменшення розділення в цьому випадку неможливо, тому що для одного стану дискретизаційне число вже мінімально й дорівнює одиниці. Крім того, неможливо зменшити розділення області, якщо коефіцієнт Грея більше r_d . Таким чином, використовуючи вирази (2.46) і (2.47), можна визначити дискретизаційне число Ω_{ij} в розглянутій області R_{ij} й крок дискретизації в тій же області.

На рис. 2.10 показана схема адаптивної дискретизації змінної x_2 , тобто тут $r_1 = 1$, r_2 – змінна.

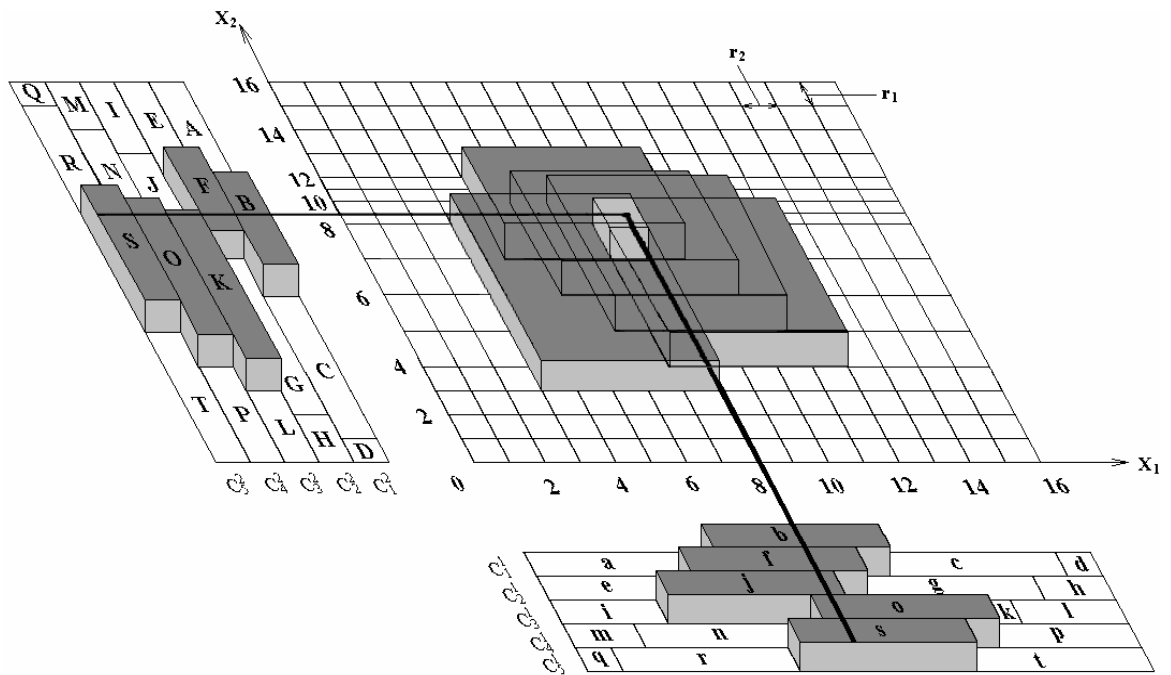


Рисунок 2.10 – Схема дискретизації СМАС після корекції

У випадку використання адаптивної дискретизації, алгоритм навчання мережі СМАС матиме такий вигляд:

- 1) Вибрати адекватне число ступенів дискретизації n .
- 2) Зробити настроювання вагових коефіцієнтів за допомогою будь-якого алгоритму навчання мережі СМАС.
- 3) Обчислити вихідні значення мережі СМАС для всіх наборів вхідних навчальних даних.
- 4) Обчислити коефіцієнти подібності Грея між реальними й бажаними вихідними сигналами за допомогою (2.45).
- 5) Скорегувати схему дискретизації вхідного простору відповідно до функції ухвалення рішення про дискретизацію (2.46).
- 6) Перерозподілити вміст пам'яті у відповідності зміною схеми квантування.
- 7) Якщо загальна помилка зросла, і швидкість навчання стала менше або пройшли всі кроки навчання, необхідно тимчасово призупинити навчання; інакше повторювати із кроку 2 поки результати не будуть задовільними.

2.1.7. Мережа Low Dimensional Based CMAC

Одним з ефективних способів мінімізації об'єму необхідної пам'яті мережі CMAC при роботі з багатомірними об'єктами є застосування запропонованих в [312] мереж Low Dimensional Based CMAC (LDB CMAC). Використання подібного підходу звільняє розроблювача від деяких проблем, що супроводжують реалізації алгоритму зворотного поширення помилки при дослідженні багатомірних функцій.

Приклад структури мережі LDB CMAC показаний на рис. 2.11, де n – кількість базових нейронних мереж CMAC, а N – число вхідних змінних. В основі такої структури лежить безліч малорозмірних (базових) нейронних мереж CMAC (наприклад, двовимірних), на які подаються всі можливі парні комбінації вхідних сигналів.

Зважені вагами z_i ($i = 1, 2, \dots, n$) виходи цих CMAC формують загальний вихід мережі

$$y(x) = \sum_{i=1}^n y_i(x) z_i, \quad (2.51)$$

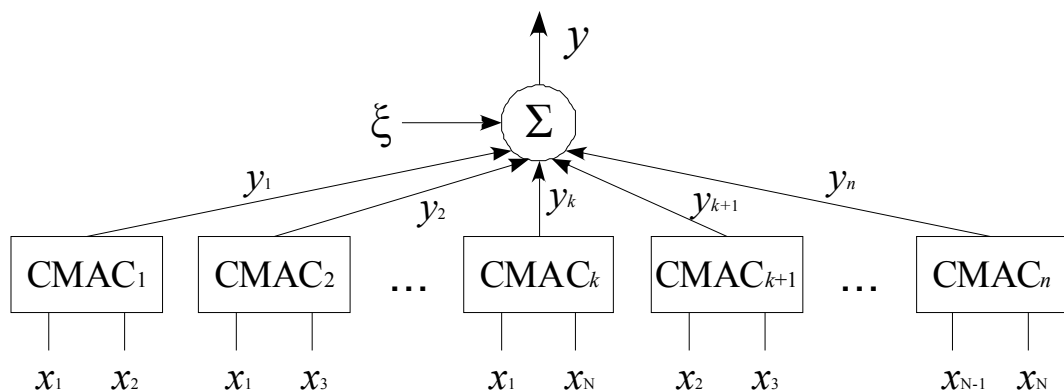


Рисунок 2.11 – Структура мережі LDB CMAC

де $y_i(x)$ – вихідне значення базової мережі $CMAC_i$; z_i – вага відповідного виходу $y_i(x)$, $i = 1, 2, \dots, n$.

Крім цього, розглянута архітектура дає можливість створення неповних структур, у яких беруть участь не всі можливі комбінації пар вхідних змінних, а лише частина з них.

Процедура навчання нейронної мережі LDB СМАС полягає в наступному:

1) Настроювання ваг z_i , наприклад, відповідно алгоритму

$$z(k+1) = z(k) + \frac{y(x) - \hat{y}(x)}{\|\mathbf{Y}(k)\|^2} \mathbf{Y}(k), \quad (2.52)$$

де $\mathbf{Y}(k) = (y_1(k), y_2(k), \dots, y_n(k))$ – вектор виходів базових мереж СМАС.

2) Настроювання параметрів базових мереж СМАС_{*i*} ($i = 1, 2, \dots, n$) за допомогою будь-якого алгоритму навчання мережі СМАС.

Слід зазначити, що на відміну від ШНМ НСМАС, розглянутої далі, LDB СМАС не вимагає використання диференційовних БФ і дозволяє застосовувати градієнтний метод навчання при виборі БФ будь-якої форми, включаючи найпростішу прямокутну.

2.1.8. Ієрархічна мережа НСМАС

Як більш ефективний спосіб рішення описаних вище проблем пропонується застосування ієрархічної структури СМАС (Hierarchical СМАС - НСМАС), що складається з декількох простіших модулів, наприклад, двовимірних СМАС з неконстантними диференційовуваними гавсівськими БФ (ГСМАС).

Запропонована НСМАС вирішує проблему підвищеної вимоги до пам'яті вихідної моделі СМАС за допомогою декомпозиції багатомірної задачі на декілька двовимірних підзадач. Можливе застосування структури, що самоорганізується, вхідного простору, заснованої на мірі ентропії Шеннона і методі пошуку золотим перерізом для визначення оптимальної схеми дискретизації вхідного простору, основою якої є розподіл вхідних навчальних даних.

У порівнянні з іншими підходами до дискретизації вхідних просторів структури, що самоорганізується, не потрібна інформація про похідну. Цей спосіб більше підходить для розв'язання задач класифікації образів, але, у свою чергу, не може бути застосован для подання дискретизації вхідного простору в випадках, що не можуть бути категоризовані, наприклад при розв'язанні задач апроксимації функцій.

Після того, як вхідний простір розбивається на дискретні області, виділення пам'яті в нейронній мережі НСМАС відбувається автоматично. Для настроювання параметрів мережі НСМАС, що самоорганізується, пропонується градієнтно-спадне навчальне правило.

Структура нейронної мережі НСМАС, що самоорганізується, ієрархічно розкривається відповідно до топології повного бінарного дерева відповідно до розмірності вхідних векторів. При цьому можливо створення надлишкових вузлів ГСМАС у випадку, коли розмірності вхідних векторів у розв'язуваній задачі не дорівнюють ступеням двійки й, відповідно, не збігаються із числом входів НСМАС. Для того щоб обминути обмеження, що накладає на розмірності вхідних векторів розв'язуваних задач запропоновано надлишкові вхідні змінні мережі НСМАС дорівнювати нулю. Але при цьому, за рахунок наявності надлишкових вузлів ГСМАС, значно збільшується об'єм необхідної пам'яті, що у свою чергу негативно впливає на швидкість і збіжність процесу навчання НСМАС.

У зв'язку із цим актуальним є розв'язання задачі пошуку мінімальної структури нейронної мережі НСМАС, що самоорганізується, яка має таку ж розмірність вхідного простору, як і розв'язувана нею задача. Для одержання мінімальної структури мережі СМАС, що самоорганізується, насамперед коротко проаналізуємо архітектуру мережі НСМАС і підхід до самоорганізації вхідного простору.

На рис. 2.12 наведено найменшу топологію нейронної мережі НСМАС, у якій кожна мережа ГСМАС містить у собі два вхідних параметри, а значення

на виході GСМАС другого рівня служать як вхідні значення для GСМАС першого рівня.

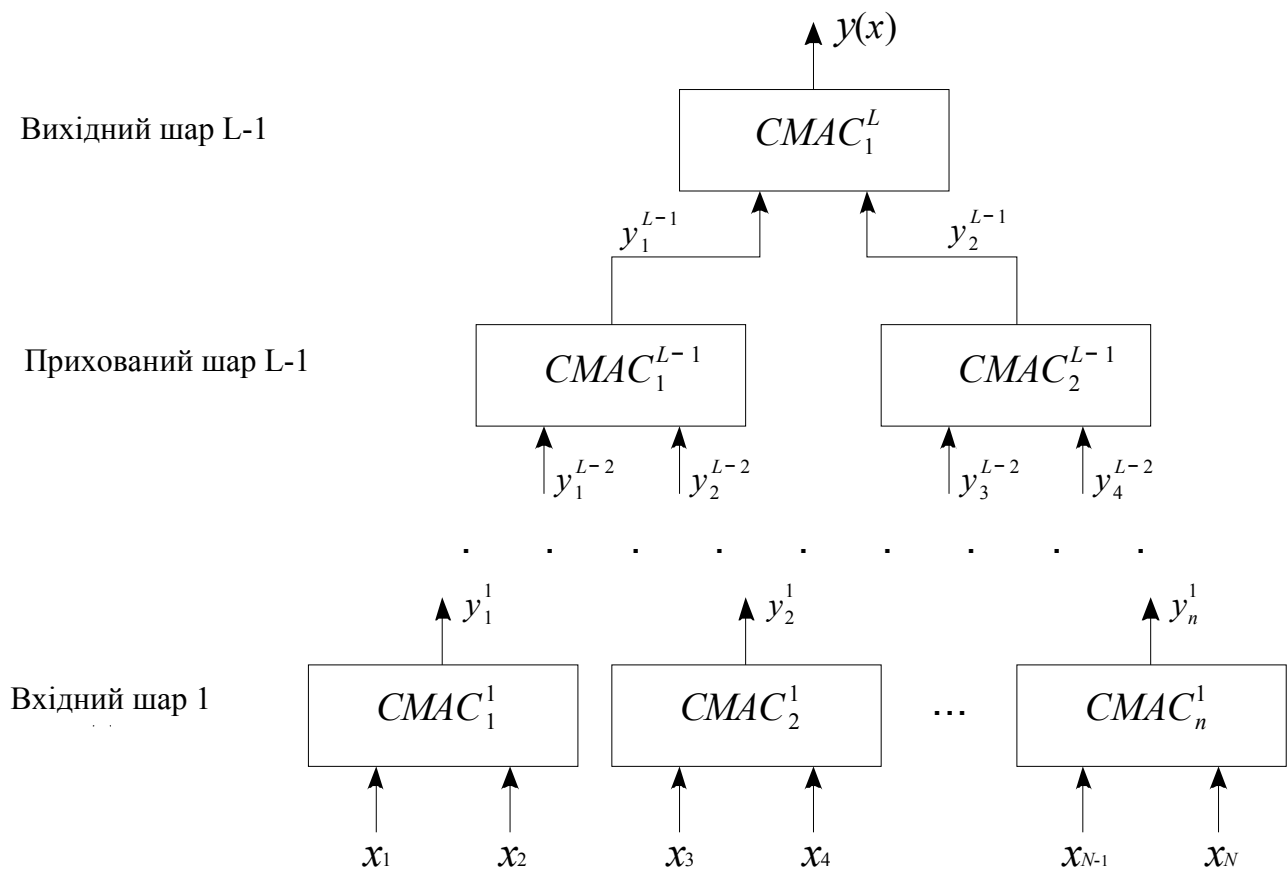


Рисунок 2.12 – Мінімальна структура ШНМ НСМАС

Тут використовуються такі позначення: x_i ($i=1, 2, \dots, N$) – i -й вхід нейронної мережі НСМАС; y_j^l ($i=1, 2, \dots, n$; $l=1, 2, \dots, L$) – вихід j -й GСМАС шару l ; n – кількість мереж GСМАС у шарі l ; $y(x)$ – вихідний сигнал НСМАС для вхідного сигналу x .

На рис. 2.13 наведено топологію НСМАС, яка враховує, що кожна мережа GСМАС містить два входи, а вихідний сигнал GСМАС першого шару є вхідним сигналом для GСМАС другого шару й т.п. Завдяки тому, що мережа НСМАС використовує топологію повного бінарного дерева, то, наприклад, задачі класифікації образів із трьома й чотирма вхідними ознаками будуть

вирішуватися за допомогою НСМАС однакових архітектур. У цьому випадку, при розв'язанні задачі із трьома вхідними вимірами буде створений один надлишковий вузол GСМАС ($GСМАС_3$), тим самим займаючи більший обсяг пам'яті.

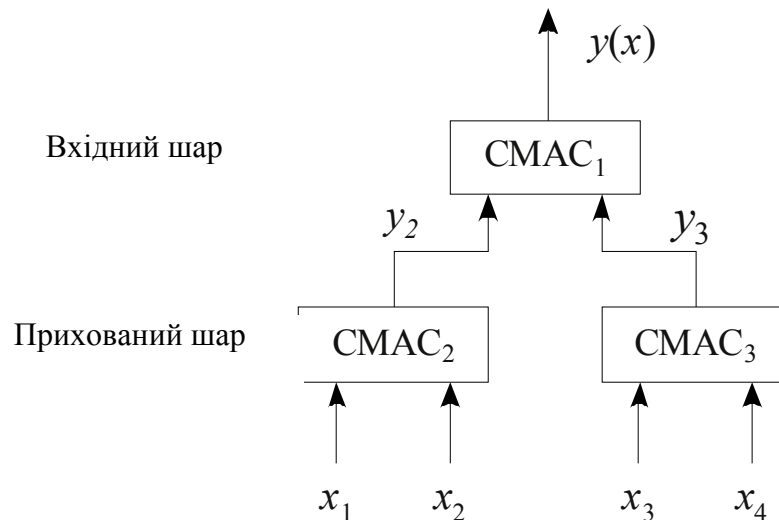


Рисунок 2.13 – Топологічна структура ШНМ НСМАС

Очевидно, що витрати пам'яті значно зростуть при розв'язанні задач із більшими розмірностями вхідного простору, відмінними від ступеня двійки. Зокрема, розв'язання задачі класифікації образів з 513 вхідними ознаками призведе до створення структури з 1024 вхідними змінними, тим самим генеруючи 511 надлишкових вузлів GСМАС.

Для виводу правила навчання мережі НСМАС розглянемо представлену на рис. 2.14 детальну архітектуру мережі HGСМАС (Hierarchical Gaussian СМАС). Тут як БФ застосовуються диференційовувані гавсівські функції. Відповідно до рис. 2.14, у процесі навчання HGСМАС мають настроюватися три види параметрів для кожної вхідної до неї двовимірної GСМАС: ваги w , радіуси σ й центри m .

При виборі диференційовуваних БФ для навчання розглянутої мережі (настроювання її параметрів) застосуємо градієнтний алгоритм зворотного розповсюдження помилки. У цьому випадку спочатку провадиться

настроювання параметрів GCMAC вихідного шару L , після чого помилка розповсюджується у зворотному напрямку від L -го шару до $(L-1)$ -го прихованого шару для настроювання параметрів GCMAC $(L-1)$ -го шару й так далі. Якщо як функціонал, що мінімізують, обрано квадратичний функціонал помилки

$$E = \frac{1}{2}(\hat{y}(x) - y(x))^2, \quad (2.53)$$

де $\hat{y}(x)$ й $y(x)$ – необхідний й реальний вихідні сигнали мережі НСМАС для вхідного сигналу x відповідно, то узагальнена процедура навчання нейронної мережі можна подати так:

1) Настроювання GCMAC вихідного шару. Настроювання параметрів мережі GCMAC вихідного шару практично нічим не відрізняється від настроювання звичайної двовходової мережі СМАС, за винятком того, що вхідними сигналами для GCMAC вихідного шару є виходи мереж GCMAC попереднього шару. Настойка здійснюється за допомогою будь-якого алгоритму навчання мережі СМАС.

2) Настроювання мереж GCMAC прихованих шарів. Для виводу алгоритма навчання мереж GCMAC у прихованому шарі з використанням алгоритма зворотного розповсюдження помилки в першу чергу необхідно

обчислити частинні похідні $\frac{\partial y_{[h/2]}^{l+1}(x)}{\partial y_h^l}$ за допомогою виразу

$$\frac{\partial y_{[h/2]}^{l+1}(x)}{\partial y_h^l} = \sum_{i=1}^{\rho} a_i(y^l) \cdot w_{hi}^l \cdot \left[\prod_{j=2}^{[h/2]+1} \Phi_i(y_j^l) \right] \cdot \frac{\partial \Phi_i(y_h^l)}{\partial y_h^l}, \quad (2.54)$$

де $h = 1, 2, \dots, n$ – номер мережі СМАС у поточному шарі; а $]\bullet[$ – позначає округлення до найближчого більшого цілого числа.

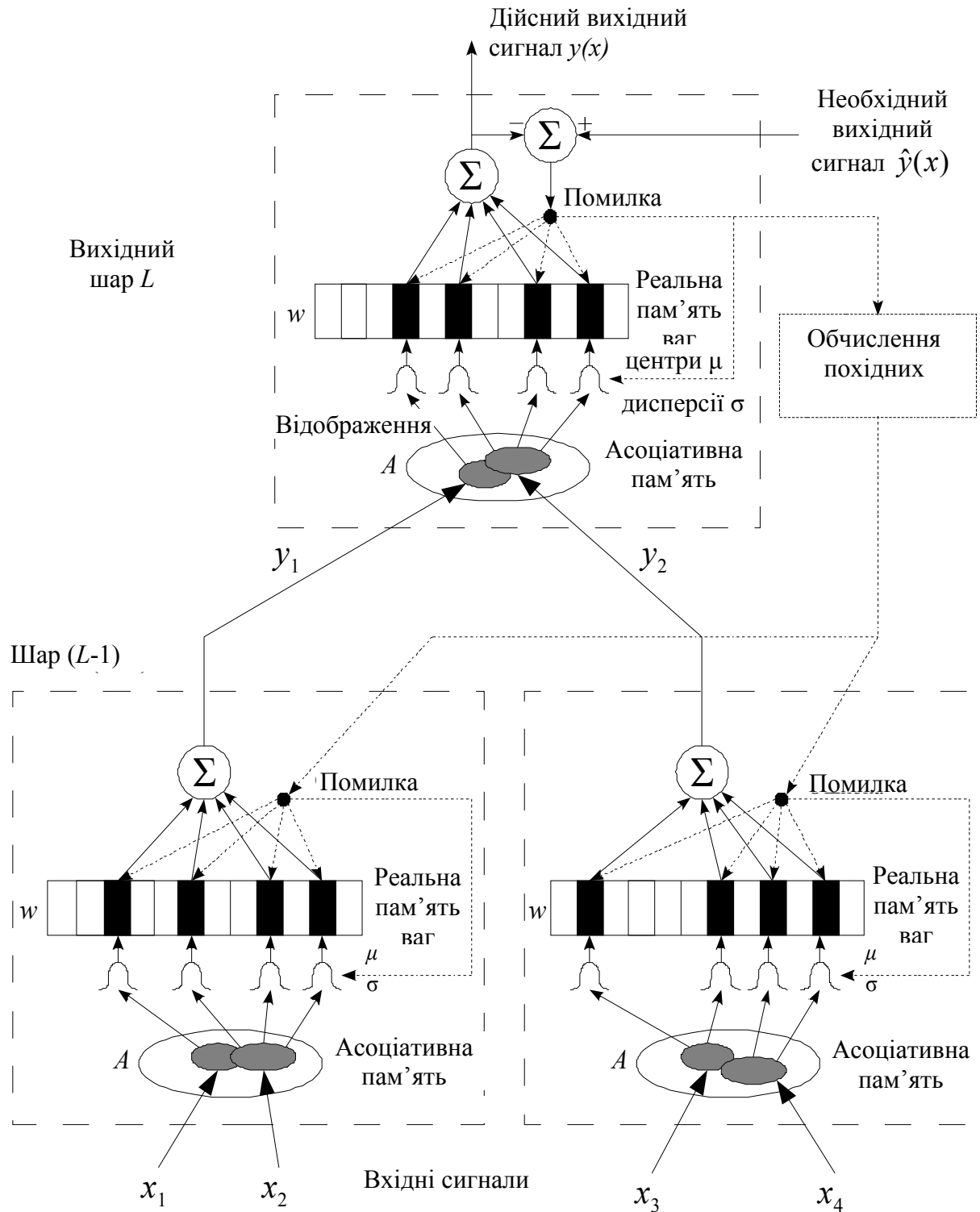


Рисунок 2.14 – Докладна архітектура ШНМ НСМАС

Далі запишемо алгоритм навчання мереж GСМАС у прихованому шарі:

$$w_h^l(k+1) = w_h^l(k) + \gamma \cdot [\hat{y}_{h/2}^{l+1}(y^l) - y_{h/2}^{l+1}(y^l)] \cdot \frac{\partial y_{h/2}^{l+1}(x)}{\partial y_h^l} \cdot \hat{O}(y^l) a(k), \quad (2.55)$$

де w_{hi}^l – значення ваги в i -ій комірці пам'яті h -ої мережі СМАС шару l ; γ – коефіцієнт навчання.

Приведена вище процедура повторюється доти, поки не буде досягнуто задане значення критерію помилки (2.53), або задане максимальне число ітерацій навчання.

2.1.9. Мінімізація структури мережі НСМАС

З метою усунення перевитрати пам'яті й поліпшення характеристик навчання НСМАС, пропонується мережа МНСМАС, що є НСМАС мінімальної конфігурації з розмірністю вхідного вектора, рівною числу вхідних змінних розв'язуваної прикладної задачі. Приводиться опис структури нейронної мережі МНСМАС, механізму самоорганізації вхідного простору, порівняння МНСМАС зі звичайною мережею, що самоорганізується, НСМАС і докладний опис алгоритму навчання нейронної мережі МНСМАС.

2.9.1 Архітектура МНСМАС. Нейронна мережа, що самоорганізується, МНСМАС, як і НСМАС, є ієрархічною структурою, що складається із двовимірних диференційовуваних мереж GСМАС, але може розширюватися довільним образом, використовуючи для масштабування топологію точного бінарного дерева. Таким чином, МНСМАС має ту ж розмірність вхідного простору, що й розв'язувана прикладна задача, тобто не містить надлишкових входів і вузлів.

На рис. 2.15 наведено кілька нейронних мереж МНСМАС з різними значеннями розмірностей вхідних векторів.

Тут x_i позначає задану вхідну послідовність ознак. Докладний опис алгоритму автоматичного призначення вхідних ознак входам нейронної мережі МНСМАС наведено далі.

Очевидно, що структура нейронної мережі МНСМАС буде збігатися зі структурою НСМАС у випадку, коли вхідні вектори розв'язуваної задачі мають у точності 2^n вимірів, де $n \in Z$. Навпроти, у випадку, коли вхідні вектори мають розмірність відмінну від 2^n , структура мережі МНСМАС буде відрізнятися від структури НСМАС.

Топології нейронних мереж МНСМАС, наведені на рис. 2.15, дозволяють вивести загальне правило автоматичного визначення структури МНСМАС. При побудові структури МНСМАС у першому шарі створюється кореневий вузол ГСМАС, позначуваний як $ГСМАС_1$. Потім, у наступному шарі створюються другий і третій вузли ГСМАС, позначувані відповідно як $ГСМАС_2$ і $ГСМАС_3$, і так далі. У структурі МНСМАС кожному вузлу ГСМАС присвоюються послідовні номери, які можуть використатися для визначення зв'язків між батьківськими й дочірніми вузлами ГСМАС у структурі МНСМАС. Кореневий вузол ГСМАС ($ГСМАС_1$) перебуває у вихідному шарі L , тоді як інші вузли розташовані в прихованих шарах $(L-1)$, $(L-2)$ і т.д.

Кількість вузлів ГСМАС, необхідна для побудови структури одиночної МНСМАС, обчислюється на основі числа вимірів вхідного вектора відповідно до виразу

$$N_{ГСМАС_s} = N - 1, \quad (2.56)$$

де $N_{ГСМАС_s}$ – кількість вузлів ГСМАС, необхідна для побудови повної структури нейронної мережі МНСМАС із числом входів N .

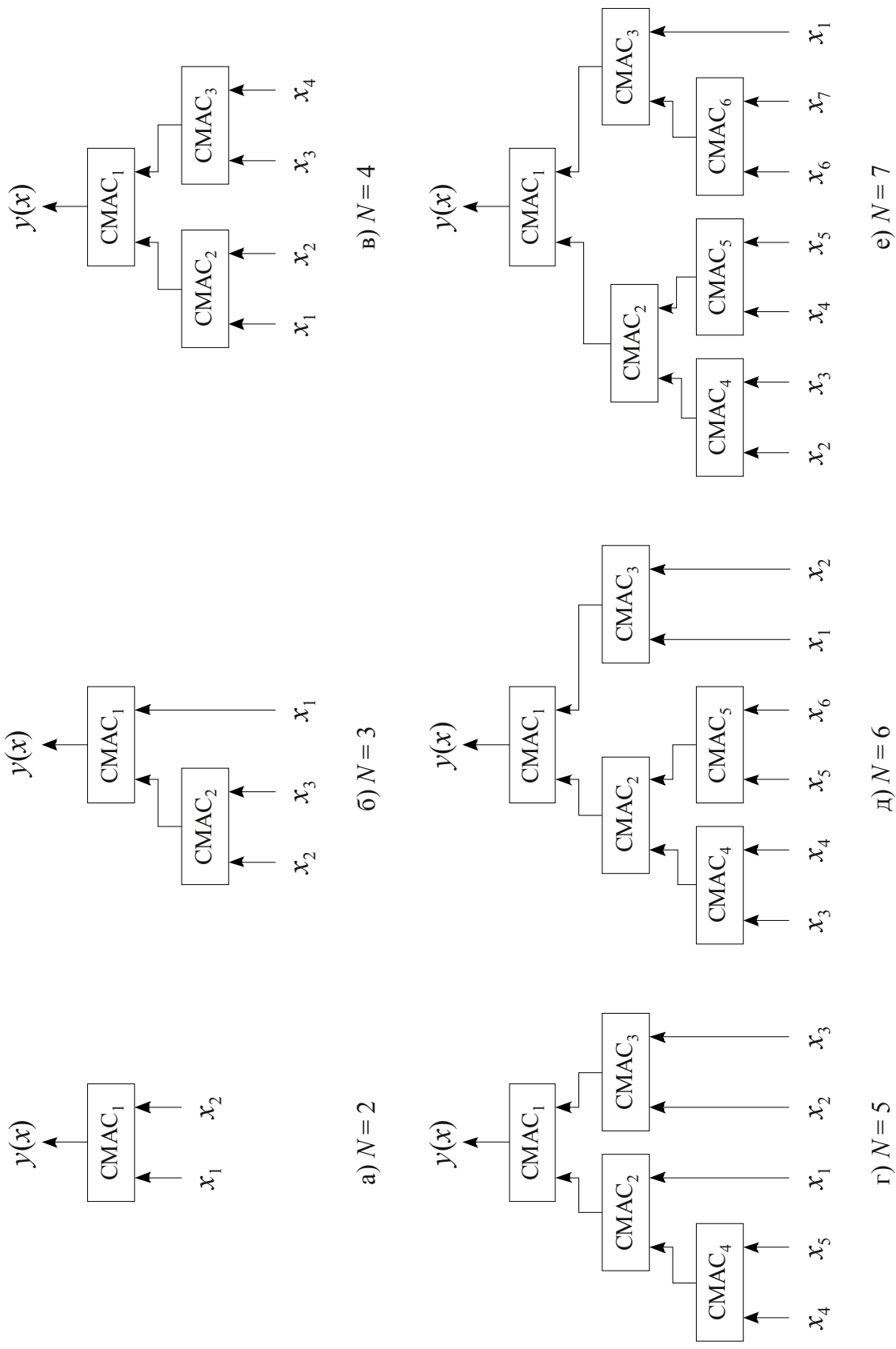


Рисунок 2.15 – Мінімальна структура ШНМ НСМАС з різною кількістю вхідних змінних

Для розв'язання задач класифікації з множиною вихідних категорій сумарна кількість вузлів GСМАС, необхідна для побудови повної структури МНСМАС, обчислюється як

$$N_{GСМАСs} = N_y \cdot (N - 1), \quad (2.57)$$

де N_y – кількість вихідних категорій у розв'язуваній задачі.

Батьківський вузол i -ої GСМАС у структурі нейронної мережі МНСМАС можна визначити, використовуючи вираз

$$P_{GСМАС_i} = \lfloor i/2 \rfloor, \quad (2.58)$$

де $P_{GСМАС_i}$ – номер вузла, що є батьківським для вузла $GСМАС_i$, $i = 2, 3, 4, \dots, N_{GСМАСs}$, $\lfloor \bullet \rfloor$ – округлення до найменшого найближчого цілого.

Дочірній вузол i -ої GСМАС обчислюється як

$$C_{GСМАС_i} = i \bmod 2, \quad (2.59)$$

де $C_{GСМАС_i} = 0$ позначає, що вузол $GСМАС_i$ є лівим дочірнім вузлом для свого батьківського вузла; а $C_{GСМАС_i} = 1$ виходить, що вузол $GСМАС_i$ є правим дочірнім вузлом для свого батьківського вузла.

Номер прихованого шару структури МНСМАС, у якому знаходиться вузол $GСМАС_i$, можна обчислити за формулою

$$L_{GСМАС_i} = L - \lceil \log_2(i) \rceil, \quad (2.60)$$

де $L_{GСМАС_i}$ – номер шару, що містить вузол $GСМАС_i$, $i = 2, 3, 4, \dots, N_{GСМАСs}$.

У пропонуваній структурі МНСМАС у випадку, коли число вхідних змінних у розв'язуваній задачі є непарним числом, структура мережі МНСМАС буде мати один вузол GСМАС, на один із входів якого подається вихід його дочірнього вузла GСМАС, а на іншій - безпосередньо одна із вхідних змінних. Визначити номер цього вузла в структурі МНСМАС можна, використовуючи вираз

$$N_U = \frac{N_{GCMACs}}{2}, \quad (2.61)$$

де N_U – номер вузла GСМАС, на входи якого буде подаватися вихід його дочірнього вузла GСМАС і вхідна змінна.

Відмітимо, що відповідно до формули (2.56), у задачах з непарним числом вхідних змінних N_{GCMACs} завжди буде числом парним.

Якщо ж розв'язувана задача має у своєму розпорядженні парне число вимірів, тоді лівий і правий входи кожного задіяного вузла GСМАС на нижньому шарі відповідають вхідним змінним розв'язуваної задачі.

Порівняння структур НСМАС і МНСМАС. Таблиця 2.2 дозволяє порівняти ШНМ НСМАС і МНСМАС з погляду витрат пам'яті, топологічної структури й підходу до присвоєння вхідних змінних розв'язуваної прикладної задачі входам нейронної мережі.

Таблиця 2.2 – Порівняння ШНМ НСМАС і МНСМАС

Елементи \ Моделі	НСМАС	МНСМАС
Загальна кількість вузлів GСМАС, необхідна для побудови топологічної структури мережі	$N_y \times (2^{\lceil \log_2 N \rceil} - 1)$	$N_y \times (N - 1)$
Топологічна структура	Структура повного бінарного дерева	Структура точного бінарного дерева

Моделі Елементи	НСМАС	МНСМАС
Алгоритм призначення вхідних змінних розв'язуваної задачі входам ШНМ	Довільний порядок призначення ознак входам листових вузлів GСМАС	Довільний порядок призначення, або пропонований метод автоматичного присвоєння вхідних ознак
Алгоритм призначення входів надлишкових листових вузлів GСМАС	Присвоєння нульових значень входам надлишкових вузлів GСМАС	Відсутність надлишкових листових вузлів GСМАС

У таблиці 2.2 прийняті такі позначення: N – кількість вхідних ознак, N_y – число вихідних категорій розв'язуваної задачі, а $\lceil \bullet \rceil$ позначає округлення убік найближчого більшого цілого.

Так як структура мережі НСМАС, що самоорганізується, розкривається на основі топології повного бінарного дерева незалежно від числа вхідних змінних, в НСМАС при розмірності вхідних векторів, відмінної від ступеня двійки, неминуче виникають надлишкові листові вузли GСМАС і надлишкові входи, на які подаються константні нульові значення. У свою чергу, МНСМАС позбавлена зазначеного недоліку. Завдяки використанню топології точного бінарного дерева витрати пам'яті для мережі МНСМАС зростають лінійно з ростом кількості вхідних змінних, у той час як витрати пам'яті НСМАС ростуть зі ступенем двійки граничного логарифма кількості вхідних вимірів.

Вхідні змінні розв'язуваної прикладної задачі можуть присвоюватися довільним входам листових вузлів GСМАС як у що самоорганізується НСМАС, так і в МНСМАС. Проте, з метою спрощення апаратної реалізації МНСМАС у цьому розділі пропонується метод автоматичного присвоєння вхідних змінних відповідним їм входам і вузлам GСМАС у топологічній структурі МНСМАС.

Самоорганізація вхідного простору МНСМАС. Формалізація порядку призначення вхідних змінних розв'язуваної задачі входам мережі МНСМАС має вирішальне значення при її практичній реалізації, зокрема, для розробки комп'ютерної програми.

Загальне правило автоматичного присвоєння вхідних ознак вхідним змінним і створення структури МНСМАС виводиться на основі аналізу властивостей набору вхідних змінних. Для виводу загального правила самоорганізації вхідного простору мережі МНСМАС скористаємося прикладами структур, наведеними на рис. 2.12.

Введемо параметри K_j й M_j , які визначають номер вузла GСМАС і вхід, якому буде привласнена j -а вхідна змінна x_j . Значення K_j й M_j визначаються відповідно за допомогою наступних виразів

$$K_j = \left\lceil \frac{j + N_{GCMACs}}{2} \right\rceil, \quad (2.62)$$

$$M_j = (j + N_{GCMACs}) \bmod 2, \quad (2.63)$$

де K_j вказує на те, що j -а вхідна змінна x_j присвоюється лівому входу K_j -го вузла $GCMAC_{K_j}$ в структурі МНСМАС у випадку, якщо $M_j = 0$, і, відповідно, правому входу вузла $GCMAC_{K_j}$, якщо $M_j = 1$.

Наведене правило є універсальним для побудови нейронних мереж МНСМАС, тому що з його допомогою можна автоматично призначати вхідні змінні розв'язуваної задачі вхідним сигналам МНСМАС як для парних розмірностей вхідних векторів, так і для непарних.

Тому що класичні ШНМ СМАС використовують рівномірну схему дискретизації вхідного простору сигналів, вони не здатні точно відобразити фактичний розподіл навчальних даних, призводячи тим самим до додаткових витрат пам'яті й зниженню продуктивності навчання нейронної мережі.

Як альтернатива також може бути використаний запропонований вище метод адаптивного настроювання схеми дискретизації, заснований на аналізі подібності Грея. На підставі інформації про схему дискретизації простору вхідних сигналів автоматично будуються вектори асоціативної адресації й фактична структура пам'яті МНСМАС.

У загальному вигляді алгоритм автоматичної побудови структури нейронної мережі, що самоорганізується, МНСМАС можна представити так:

1) визначення схеми дискретизації вхідного простору сигналів за допомогою методу вхідних просторів, що самоорганізуються, заснованого на аналізі міри ентропії Шеннона набору навчальних даних, або метод, заснований на аналізі подібності Грея;

2) автоматична побудова структури МНСМАС із використанням ознак K_j і M_j , що обчислюють за формулами (2.62) і (2.63).

Слід зазначити, що методи самоорганізації вхідного простору застосовуються тільки для листових вузлів GСМАС, що знаходяться у вхідному шарі, тому що в цьому випадку вони ґрунтуються на апріорно відомій інформації про розподіл вхідних навчальних даних. Для інших вузлів GСМАС, розташованих у прихованих і вихідних шарах структури МНСМАС, використовується схема рівномірної дискретизації вхідного простору, тому що розподіл вхідних сигналів вузлів GСМАС у прихованих шарах постійно змінюється в процесі навчання, і інформація про кінцевий розподіл може бути отримана лише по закінченню процесу навчання нейронної мережі.

Правило навчання ШНМ МНСМАС. Для спрощення виводу правила навчання мережі МНСМАС розглянемо приклад навчання МНСМАС із п'ятьма вхідними змінними, топологічна структура якої наведена на рис. 2.16.

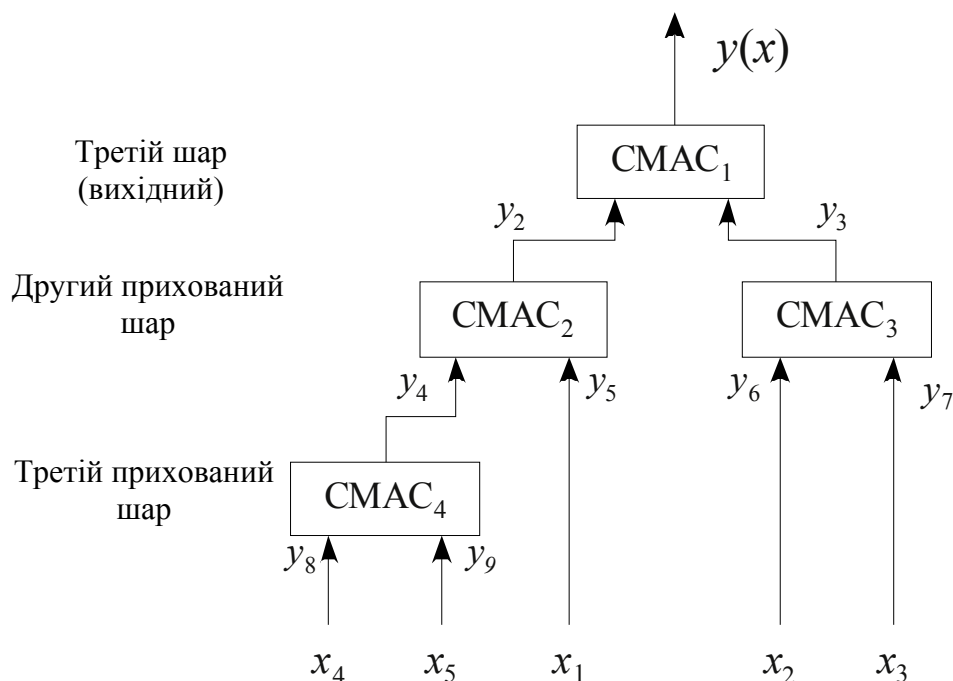


Рисунок 2.16 – Топологічна структура МНСМАС з п'ятьма вхідними змінними

Дана мережа має п'ять вхідних сигнали, позначені як x_i ($i = 1, 2, \dots, 5$) y_j ($j = 1, 2, 3, 4$) є виходом j -го вузла $GCMAC_j$, при цьому y_2 , y_3 і y_4 – виходи вузлів $GCMAC$ прихованого шару, а y_1 – вихід МНСМАС. Крім того, для узагальнення й спрощення процедури виводу навчальних правил, вхідні ознаки x_i ($i = 1, 2, \dots, 5$) також позначимо як y_j , де $j = 5, 6, \dots, 9$. Виведемо навчальне правило на підставі методу градієнтного спуску й властивості диференційовності вузлових $GCMAC$.

Тому що $GCMAC_1$ L -го шару є вихідним вузлом, і його вихідне значення є виходом мережі МНСМАС, те для настроювання вагових коефіцієнтів $GCMAC_1$ застосуємо правило навчання звичайної нейронної мережі $GCMAC$ [308]. У якості функції вартості, що мінімізують, виберемо квадратичний функціонал помилки (2.53).

У кожному вузлі $GCMAC$ настроюванню підлягають три параметри, а саме: ваги w , радіуси σ й центри БФ m . Якщо на якій-небудь ітерації процесу навчання фактичні й шукане вихідні значення мережі МНСМАС

різні, то в першу чергу настраюються параметри $GCMAC_1$ у вихідному шарі. Далі на підставі інформації про похідні, помилка розповсюджується зворотно по дереву МНСМАС від першого шару до другого, третього й наступного шарів для корекції параметрів $GCMAC_2$, $GCMAC_3$ і $GCMAC_4$.

Алгоритм навчання для наведеної на рис. 2.16 п'ятивходової нейронної мережі МНСМАС у загальному вигляді може бути представлений у такий спосіб.

Крок 1 – Корекція $GCMAC_1$ вихідного шару. Вагові коефіцієнти w_j в пам'яті $GCMAC_1$ настраюються по формулі

$$w_j(k+1) = w_j(k) - \frac{\alpha}{Ne} \frac{\partial E}{\partial w_j} = \frac{\alpha}{Ne} \cdot [\hat{y}_1(x) - y_1(x)] \cdot a_j(x) \cdot \prod_{i=2}^3 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}}\right)^2}, \quad (2.64)$$

де Δw_j – це значення виправлення до значення j -го вагового елемента w_j в пам'яті $GCMAC_1$; m_{jk} – центр k -го виміру гавсівської БФ в j -му відображеному гіперкубі, k визначається за допомогою i : якщо $i=2$, тоді $k=1$, якщо $i=3$, то $k=2$; σ_{jk} – це радіус k -го виміру гавсівської БФ в j -му відображеному гіперкубі; α – коефіцієнт швидкості навчання, $a_j(x)$ є j -м елементом вектора асоціативного вибору пам'яті для окремого вхідного стану x ; а Ne – це номер відображеного гіперкуба для вхідного стану x .

Настроювання радіусів БФ здійснюються в такий спосіб:

$$\begin{aligned} \Delta \sigma_{ji} &= -\frac{\alpha}{Ne} \cdot \frac{\partial E}{\partial \sigma_{ji}} = \\ &= \frac{\alpha}{Ne} \cdot [\hat{y}_1(x) - y_1(x)] \cdot a_j(x) \cdot w_j \cdot \left[\prod_{i=2}^3 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}}\right)^2} \right] \cdot \frac{2(y_i - m_{jk})^2}{\sigma_{jk}^3}, \end{aligned} \quad (2.65)$$

де $\Delta \sigma_{ji}$ – коригувальне виправлення до i -го виміру радіуса σ_{ji} БФ для j -го відображеного гіперкуба вхідного стану x в $GCMAC_1$

У свою чергу, центри m_{jk} БФ $GCMAC_1$ настроюються відповідно до правила

$$\begin{aligned} \Delta m_{ji} &= -\frac{\alpha}{Ne} \cdot \frac{\partial E}{\partial m_{ji}} = \\ &= \frac{\alpha}{Ne} \cdot [\hat{y}_1(x) - y_1(x)] \cdot a_j(x) \cdot w_j \cdot \left[\prod_{i=2}^3 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}}\right)^2} \right] \cdot \frac{2(y_i - m_{jk})}{\sigma_{jk}^2}, \end{aligned} \quad (2.66)$$

де Δm_{ji} – значення виправлення i -го виміру центра m_{ji} БФ $GCMAC_1$ для j -го відображеного гіперкуба для вхідного стану x .

Крок 2 – Корекція $GCMAC_2$ й $GCMAC_3$ перший прихований шар. Оскільки інформація про оптимальні значення $\hat{y}_2(x)$ й $\hat{y}_3(x)$ не доступна, то для виводу навчальних правил настроювання параметрів $GCMAC_2$ і $GCMAC_3$ в першому прихованому шарі $(L-1)$ спочатку необхідно обчислити похідні $\frac{\partial y_1(x)}{\partial y_2}$ і $\frac{\partial y_1(x)}{\partial y_3}$ як

$$\frac{\partial y_1(x)}{\partial y_2} = \sum_{j=1}^{N_h} a_j(x) \cdot w_j \cdot \left[\prod_{i=2}^3 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}}\right)^2} \right] \cdot \frac{-2(y_2 - m_{j1})}{\sigma_{j1}^2} \quad (2.67)$$

та

$$\frac{\partial y_1(x)}{\partial y_3} = \sum_{j=1}^{N_h} a_j(x) \cdot w_j \cdot \left[\prod_{i=2}^3 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}}\right)^2} \right] \cdot \frac{-2(y_3 - m_{j2})}{\sigma_{j2}^2} \quad (2.68)$$

відповідно.

У наведених вище рівняннях w_j позначає j -й ваговий коефіцієнт у пам'яті $GCMAC_1$; m_{jk} є центром k -го виміру БФ для j -го відображеного гіперкуба вхідного стану x в $GCMAC_1$; σ_{jk} – радіус k -го виміру БФ для j -

го відображеного гіперкуба вхідного стану x в $GCMAC_1$; N_h – це спільний фактичний розмір пам'яті $GCMAC_1$; а k визначається за допомогою i : якщо $i = 2$, то $k = 1$, якщо $i = 3$, то $k = 2$.

Далі, правила настроювання ваг w_j , радіусів σ_{ji} і центрів m_{ji} БФ $GCMAC_2$ у першому прихованому шарі $(L - 1)$ МНСМАС виводяться в такий спосіб:

$$\begin{aligned} \Delta w_j &= -\frac{\alpha}{Ne} \cdot \frac{\partial E}{\partial w_j} = \\ &= \frac{\alpha}{Ne} \cdot [\hat{y}_1(x) - y_1(x)] \cdot \left(\frac{\partial y_1(x)}{\partial y_2} \right) \cdot a_j(x) \cdot \left[\prod_{i=4}^5 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}} \right)^2} \right] \end{aligned} \quad (2.69)$$

$$\begin{aligned} \Delta \sigma_{ji} &= -\frac{\alpha}{Ne} \cdot \frac{\partial E}{\partial \sigma_{ji}} = \\ &= \frac{\alpha}{Ne} \cdot [\hat{y}_1(x) - y_1(x)] \cdot \left(\frac{\partial y_1(x)}{\partial y_2} \right) \cdot a_j(x) \cdot w_j \cdot \left[\prod_{i=4}^5 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}} \right)^2} \right] \cdot \frac{2(y_i - m_{jk})^2}{\sigma_{jk}^3}, \end{aligned} \quad (2.70)$$

$$\begin{aligned} \Delta m_{ji} &= -\frac{\alpha}{Ne} \cdot \frac{\partial E}{\partial m_{ji}} = \\ &= \frac{\alpha}{Ne} \cdot [\hat{y}_1(x) - y_1(x)] \cdot \left(\frac{\partial y_1(x)}{\partial y_2} \right) \cdot a_j(x) \cdot w_j \cdot \left[\prod_{i=4}^5 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}} \right)^2} \right] \cdot \frac{2(y_i - m_{jk})}{\sigma_{jk}^2}. \end{aligned} \quad (2.71)$$

У рівностях (2.69), (2.70) і (2.71) використані символічні позначення аналогічні прийнятим у виразах (2.64), (2.65) і (2.66), ваги w_j , радіуси σ_{jk} й радіуси σ_{ji} належать вузлу $GCMAC_2$, а k може бути визначене за допомогою i так: якщо $i = 4$, то $k = 1$, якщо $i = 5$, то $k = 2$.

За аналогією на підставі обчислених похідних $\frac{\partial y_1(x)}{\partial y_3}$ виводиться навчальне правило для настроювання $GCMAC_3$ в першому прихованому шарі $(L - 1)$ МНСМАС.

Крок 3 – Корекція $GCMAC_4$ другого прихованого шару. Правило настроювання параметрів вузла $GCMAC_4$ другого прихованого шару ($L - 2$) аналогічно алгоритму настроювання вузлів $GCMAC_3$ і $GCMAC_3$ шаруючи ($L - 1$). Але тепер необхідна інформація про похідну $\frac{\partial y_1(x)}{\partial y_4}$, що може бути

знайдена за допомогою виразу

$$\frac{\partial y_1(x)}{\partial y_4} = \frac{\partial y_1(x)}{\partial y_2} \cdot \frac{\partial y_2(x)}{\partial y_4}, \quad (2.72a)$$

де

$$\frac{\partial y_2(x)}{\partial y_4} = \sum_{j=1}^{N_h} a_j(x) \cdot w_j \cdot \left[\prod_{i=4}^5 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}}\right)^2} \right] \cdot \frac{-2(y_4 - m_{j1})}{\sigma_{j1}^2}, \quad (2.72b)$$

де, у свою чергу, w_j позначає j -й ваговий коефіцієнт у пам'яті $GCMAC_2$; m_{jk} – центр k -го виміру БФ для j -го відображеного гіперкуба вхідного стану x в $GCMAC_2$; σ_{jk} – радіус k -го виміру БФ для j -го відображеного гіперкуба вхідного стану x в $GCMAC_2$; N_h – спільний фактичний розмір пам'яті $GCMAC_2$; а k – індекс, що визначається за допомогою i : якщо $i = 4$, то $k = 1$, якщо $i = 5$, то $k = 2$.

Отримавши похідні $\frac{\partial y_1(x)}{\partial y_2}$ й $\frac{\partial y_2(x)}{\partial y_4}$, нескладно вивести формули для настроювання ваг w_j , радіусів σ_{ji} і центрів m_{ji} БФ $GCMAC_4$ у другому прихованому шарі ($L - 2$) структури МНСМАС. У цьому випадку градієнт помилки між шуканим і фактичним виходами розповсюджується у зворотному напрямку від виходу мережі МНСМАС y_1 до першого прихованого шару ($L - 1$) згідно виразу (2.67), після чого аналогічним образом помилка розповсюджується до другого прихованого шару ($L - 2$)

відповідно до виразів (2.72a) і (2.72б). У результаті всіх підстановок виразу для настроювання параметрів вузла $GCMAC_4$ приймають вигляд

$$\begin{aligned}\Delta w_j &= -\frac{\alpha}{Ne} \cdot \frac{\partial E}{\partial w_j} = \\ &= \frac{\alpha}{Ne} \cdot [\hat{y}_1(x) - y_1(x)] \cdot \left(\frac{\partial y_1(x)}{\partial y_2} \right) \cdot \left(\frac{\partial y_2(x)}{\partial y_4} \right) \cdot a_j(x) \cdot \left[\prod_{i=4}^5 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}} \right)^2} \right], \\ \Delta \sigma_{ji} &= -\frac{\alpha}{Ne} \cdot \frac{\partial E}{\partial \sigma_{ji}} = \frac{\alpha}{Ne} \cdot [\hat{y}_1(x) - y_1(x)] \cdot \left(\frac{\partial y_1(x)}{\partial y_2} \right) \cdot \left(\frac{\partial y_2(x)}{\partial y_4} \right) \times \\ &\quad \times a_j(x) \cdot w_j \cdot \left[\prod_{i=4}^5 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}} \right)^2} \right] \cdot \frac{2(y_i - m_{jk})^2}{\sigma_{jk}^3}, \\ \Delta m_{ji} &= -\frac{\alpha}{Ne} \cdot \frac{\partial E}{\partial m_{ji}} = \frac{\alpha}{Ne} \cdot [\hat{y}_1(x) - y_1(x)] \cdot \left(\frac{\partial y_1(x)}{\partial y_2} \right) \cdot \left(\frac{\partial y_2(x)}{\partial y_4} \right) \times \\ &\quad \times a_j(x) \cdot w_j \cdot \left[\prod_{i=4}^5 e^{-\left(\frac{y_i - m_{jk}}{\sigma_{jk}} \right)^2} \right] \cdot \frac{2(y_i - m_{jk})}{\sigma_{jk}^2}.\end{aligned}$$

У останніх рівняннях використані символічні позначення, подібні прийнятим у виразах (2.64), (2.65), (2.66), з тією лише відмінністю, що ваги w_j , радіуси σ_{jk} й радіуси σ_{ji} належать вузлу $GCMAC_4$, а k визначається за допомогою i : якщо $i = 8$, то $k = 1$, якщо $i = 9$, то $k = 2$.

Отже, вирази (2.69), (2.70) і (2.71) показують, що помилка передається зворотно з $GCMAC_1$ у вихідному шарі до вузлів $GCMAC$ першого прихованого шару відповідно до інформації про похідні $\frac{\partial y_1(x)}{\partial y_2}$ і $\frac{\partial y_1(x)}{\partial y_3}$. І далі, останні рівняння показують, що помилка також розповсюджується назад з $GCMAC_1$ у вихідному шарі до вузлів $GCMAC$ другого прихованого шару відповідно до обчислених похідними $\frac{\partial y_1(x)}{\partial y_2}$ і $\frac{\partial y_1(x)}{\partial y_3}$. Отже, представляється можливим поширити це правило навчання на довільну мережу МНСМАС, що має будь-яку кількість вузлів і вхідних змінних.

2.2 Розробка структур обчислювачів на базі генераторів сплайн функцій

2.2.1 Обчислювачі на базі ФП, що реалізують сплайн однієї змінної

Цей підрозділ присвячений схемній реалізації функціональних перетворювачів, що відтворюють Q -сплайни, описані в розділі вище. Особлива увага приділена перетворювачам двох змінних, через недостатність попередніх розробок перетворювачів, що відтворюють лінійний Q -сплайн на трикутних підобластях апроксимації.

У цьому підрозділі описано програми, що реалізують алгоритми розрахунку Q -сплайнів однієї і двох змінних. Наведено результати розрахунків для сплайнів на трикутних підобластях апроксимації. Для можливості порівняння результатів розроблено програму, що здійснює розрахунок параметрів лінійного Q -сплайна на прямокутних підобластях, наведено результати її розрахунків. Виконано порівняльний аналіз результатів апроксимації різними сплайнами.

Найбільше поширення серед ФП однієї змінної одержали перетворювачі, що відтворюють кусково-лінійну функцію – сплайн першого степеня. Відтворена залежність має такий вигляд:

$$S_i(x) = q_i^{[0]} + q_i^{[1]}(x - x_i). \quad (2.73)$$

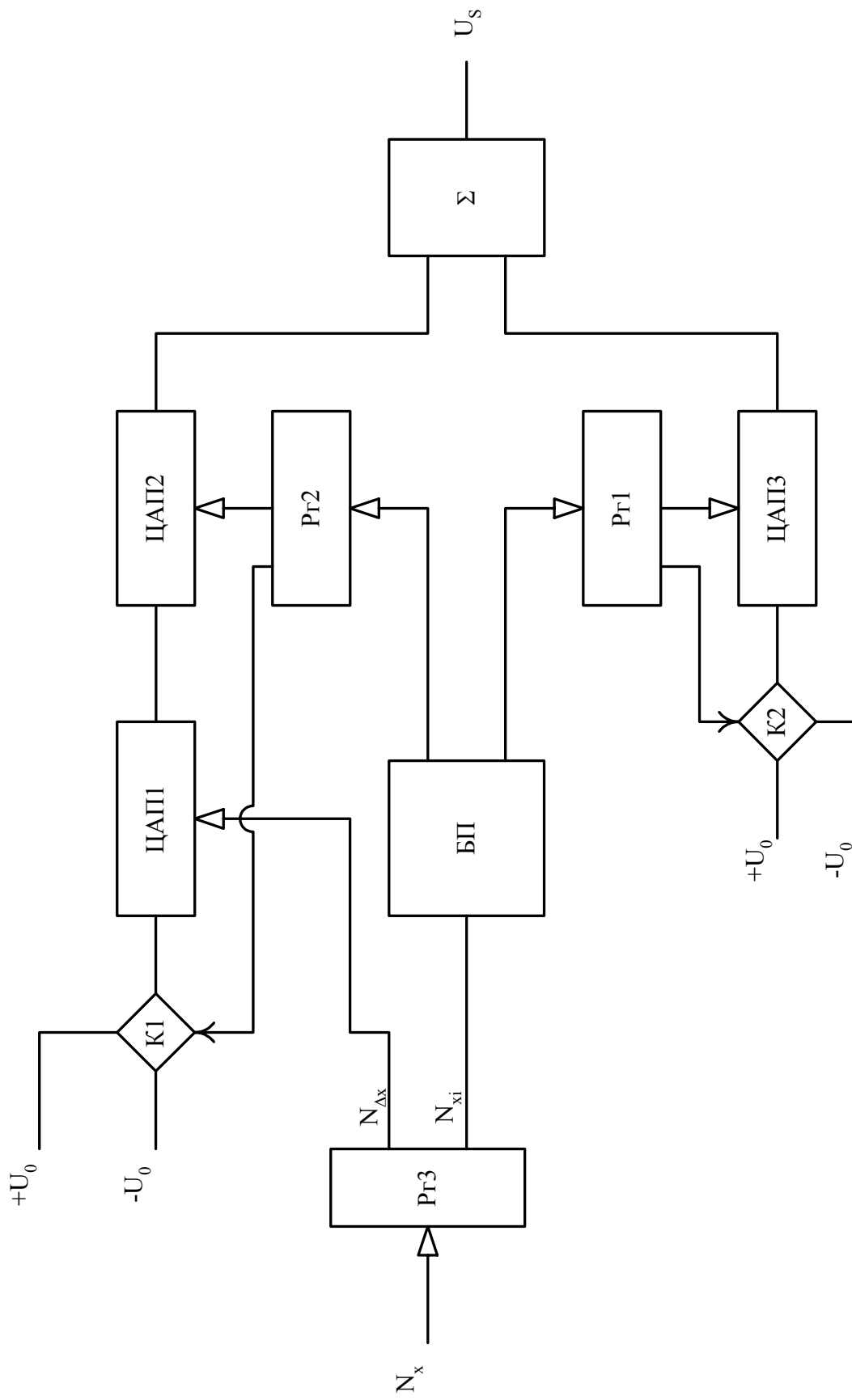
Параметри апроксимуючого сплайна $q_i^{[0]}$ і $q_i^{[1]}$ подають у цифровому вигляді і зберігаються в цифровому блоці пам'яті. Збільшення аргументу $\Delta x = (x - x_i)$ моделюється молодшими розрядами коду N_x , що є кодом аргументу x . Старші розряди коду N_x є кодом вузлової точки x_i і служать адресним словом для вибірки з блоку пам'яті параметрів апроксимації, для операції множення зручно використовувати аналоговий перетворювач, що множить, та одночасно з операцією множення робить перетворення цифрових кодів в аналогову величину.

З вищесказаного впливає структура найпростішого перетворювача, який відтворює сплайн першого степеня з квантованими параметрами, що наведена на рис. 2.17.

ЦАП1 служить для перетворення цифрового коду $N\Delta x$ – збільшення аргументу – в аналогову величину: напругу $U\Delta x$. При цьому на аналоговий вхід ЦАП1 подається опорна напруга позитивної $+U_0$, чи негативної полярності $-U_0$. Через електронний ключ K_1 , що керується знаковим розрядом коду параметра $q_i^{[1]}$, обраного з БП і поміщеного в регістр R_2 для оперативного збереження. Якщо знаковий розряд дорівнює "0", то підключається позитивна опорна напруга $+U_0$, якщо ж знаковий розряд дорівнює "1", то негативна $-U_0$. Отже, на виході ЦАП1 формується напруга, що відповідає величині $\text{sgn}(\tilde{q}_i^{[1]})\Delta x$.

Ця напруга надходить на аналоговий вхід ЦАП2, де виробляється множення на код параметра $|\tilde{q}_i^{[1]}|$. На виході ЦАП2 формується напруга, що відповідає $\text{sgn}(\tilde{q}_i^{[1]})|\tilde{q}_i^{[1]}|\Delta x$, тобто другому доданку у виразі (2.73).

На ЦАП3 формується перший доданок як добуток коду $|\tilde{q}_i^{[0]}|$ і напруги відповідної полярності, підключенням якого через ключ K_2 керує знаковий розряд коду $N_{q_i^{[0]}}$ з регістра R_1 , поміщеного туди для оперативного збереження після вибірки з блоку пам'яті.

Рисунок 2.17 – Структурна схема ФП для відтворення лінійного Q -сплайна

Напруга з виходів ЦАП2 і ЦАП3 надходить на входи суматора Σ . На його виході формується напруга, що відповідає значенню сплайна в точці x , обумовленої вхідним кодом N_x .

Загальним для більшості пристроїв подібного типу є збереження кодів параметрів у цифровому блоці пам'яті і відтворення залежності виду (2.73) або

$$\tilde{S}_i(x) = \tilde{q}_i^{[0]} + \tilde{q}_i^{[1]}(x - x_i),$$

яку легко отримати з (2.73).

Для розрахунку квантованих параметрів апроксимації для ФП, що реалізують кусково-лінійну апроксимацію функції з цифровим поданням і збереженням параметрів, розроблено програму *SPLIT*, що функціонує відповідно до алгоритму, описаному в підрозділі 2.2.

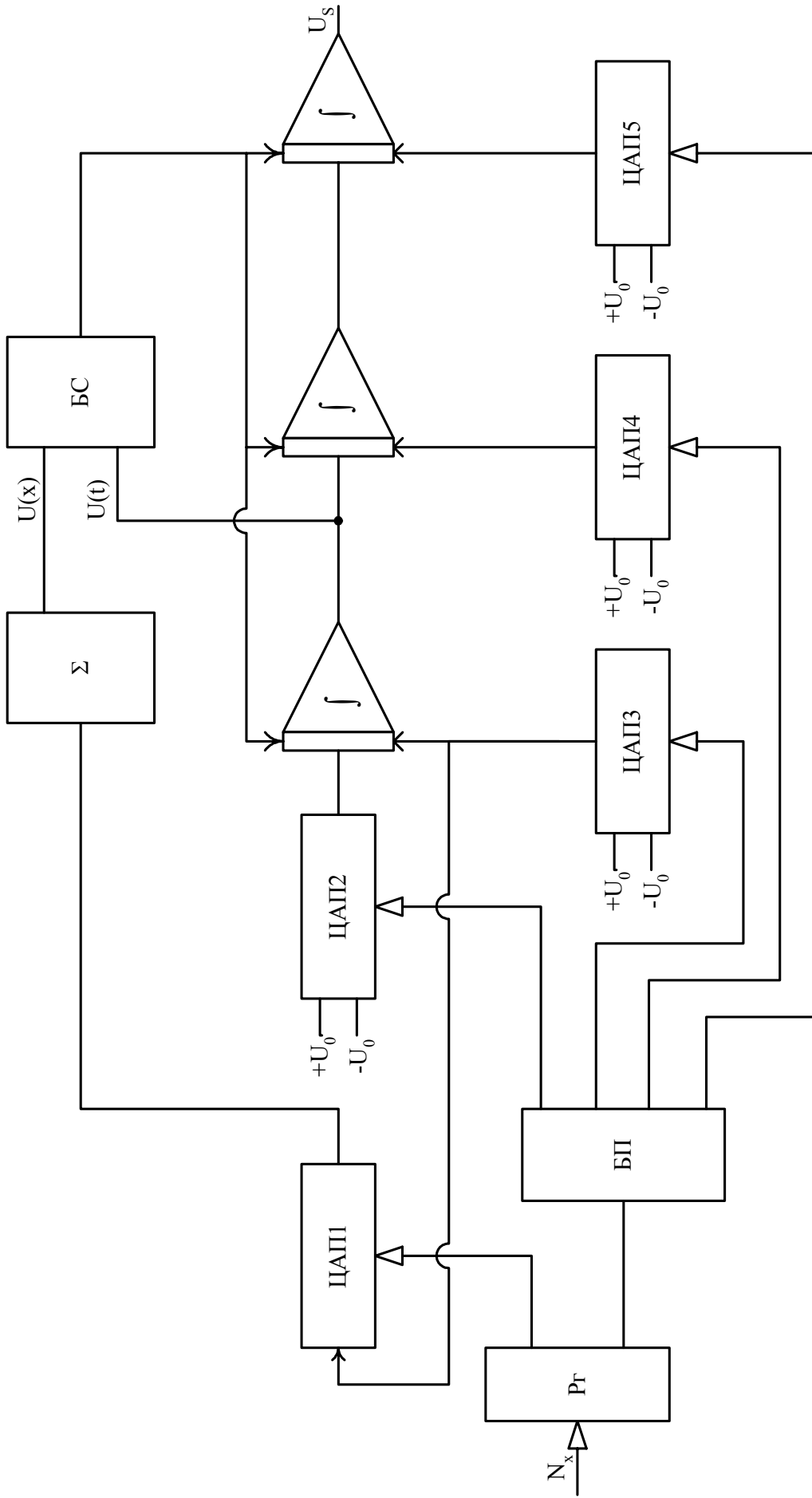
Вихідні дані програми містять: таблицю квантованих параметрів сплайна і їхніх цифрових кодів, фактичну похибку апроксимації у відсотках; кількість підобластей апроксимації – N . Крім того при $L2=1$ серед вихідних даних програми може з'явитися повідомлення про похибку обчислення другої похідної. З появою такого повідомлення і відсутності таблиці квантованих параметрів сплайна необхідно змінити спосіб обчислення другої похідної.

Набагато більш високу точність і, головне, гладкість апроксимуючої кривої дозволяють отримати сплайни більш високого порядку, ніж перший.

На основі структур генераторів сплайн-функцій і методу відстеження вхідної змінної розроблено структуру функціонального перетворювача, наведена на рис. 2.8. Запропонований перетворювач відтворює функцію довільного аргументу x , а не функцію часу, здійснює кусково-нелінійну апроксимацію сплайном, а не узагальненим поліномом.

Розглянемо роботу пристрою.

Код вхідного аргументу N_x надходить на вхідний регістр R_g для оперативного збереження. Старші розряди N_{xi} цього коду є адресним словом для вибірки з цифрового блоку пам'яті БП параметрів апроксимації. Молодші розряди $N_{\Delta x}$ коду аргументу моделюють збільшення аргументу $(x-x_i)$.

Рисунок 2.8 – Структурна схема ФП для відтворення Q -сплайна третього степеня

Цифрові коди параметрів апроксимації з блоку пам'яті надходять на цифрові входи цифро-аналогових перетворювачів ЦАП2–ЦАП5, на виходах яких формуються такі величини напруг: на виході ЦАП2 – напруга, що відповідає $b^{[n]} = (-1)^n n!q_j^{[n]}$, а на виходах ЦАП3–ЦАП5 – напруги, що відповідають початковим умовам диференціального рівняння, що розв'язується схемою: $b^{[i]} = (-1)^i i!q_j^{[i]}$, де $q_j^{[i]}$ – параметри апроксимації в j -й підобласті, цифрові коди яких зберігаються в БП; n – порядок відтвореного сплайна, що визначається кількістю інтеграторів.

На виході лівого за схемою інтегратора формується функція часу вигляду: $U(t) = 3!q_j^{[3]}t + 2q_j^{[2]}$.

Застосування ЦАП1 і суматора дозволяє сформувати напругу, що відповідає: $S(x) = 3!q_j^{[3]}x + 2q_j^{[2]}$.

Обидві ці напруги надходять на блок порівняння БС. Якщо вони не рівні, то на виході БС формується сигнал "Пуск", що дозволяє інтегрування. Коли ж напруги виявляться рівними, на виході БС формується сигнал "Зупинка", при цьому на виході перетворювача – напруга відповідає значенню сплайна в точці, що задана вхідним кодом на регістрі Rг:

$$S^{[3]}(x) = \sum_{\alpha=0}^3 q_j^{[\alpha]}(x - x_j)^\alpha.$$

Отже, запропонований пристрій дозволяє відтворювати апроксимуючий сплайн n -го степеня (на рис. 2.18 наведено схему для відтворення кубічного сплайна) від довільного аргументу.

Для розрахунку квантованих параметрів апроксимації параболічним чи кубічним сплайном за методикою, що викладена в розділі, розроблено програму *SPL23*.

Структурно програма *SPL23* складається з основної програми, чотирьох

стандартних підпрограм і підпрограми *КВАНТ* для квантування за рівнем коефіцієнтів сплайна.

Даними для програми є така інформація:

- границі області визначення функції $f(x)$ – точки a і b ;
- припустима відносна приведена похибка апроксимації ε у відсотках;
- кількість перевірних точок M ;
- масив $MRK(4)$ розрядностей коефіцієнтів, у якому містяться значення $l^{[\alpha]}$, $\alpha = \overline{0,3}$ (якщо $\beta=2$, то $l^{[3]}$ не впливає на обчислювальний процес);
- ознаки $L1$ і $L2$.

Ознака $L1$ приймає одне з таких значень: якщо $L1=1$, то виконується розрахунок коефіцієнтів параболічного і кубічного сплайнів; якщо $L1=2$, то розраховується тільки параболічний сплайн; якщо $L1=3$, то розраховується тільки кубічний сплайн.

Ознака $L2$ вказує на спосіб обчислення крайових умов: при $L2=1$ значення $f'(a)$ і $f'(b)$ визначаються методом чисельного диференціювання, при $L2=0$ визначення крайових умов відбувається безпосередньо за формулою першої похідної, що має бути задана в програмі аналітично.

У залежності від кількості перевірних точок визначається розмірність масивів: $SA(M+5)$ – масив коефіцієнтів сплайна; $KA(M)$ – масив цифрових кодів коефіцієнтів; $X(M/4+1)$ – масив вузлів інтерполяції (сітка ΔK); $Y(M/4+1)$ – масив значень функції у вузлах інтерполяції; $XSPL(M/4)$ – допоміжний масив вузлів сплайна, використовуваний у підпрограмі $SPLG2$; $XP(M)$ – масив точок перевірки умови (2.20)(сітка ΔM); $F(M)$ – масив значень сплайна в точках перевірки.

Вихідні дані програми подано у вигляді таблиці квантованих коефіцієнтів сплайна $Q(0)$, $Q(1)$, $Q(2)$, $Q(3)$ разом з десятковими номерами рівнів квантування у функції номера інтервалу апроксимації. У верхній частині таблиці друкується фактична похибка апроксимації РМ. Тут же друкується ознака результатів обчислень: якщо $KEY=0$, необхідна точність не може бути

досягнута при заданих вихідних даних; якщо $KEY=1$, то точність не може бути досягнута при заданих розрядностях коефіцієнтів; якщо $KEY=2$, то це вказує, на досягнення необхідної точності при даних розрядностях (успішний результат).

Для контролю правильності обчислень у програмі передбачено при $L2=1$ появу повідомлення про перевищення припустимого значення похибок обчислення похідних у точках a і b . З появою такого повідомлення і відсутності таблиці вихідних даних необхідно задати формулу для $f'(x)$ за допомогою оператора-функції і змінити значення ознаки $L2$ на 0.

2.2.2 Обчислювачі на базі ФП, що реалізують сплайн двох змінних

Реалізація сплайна на прямокутних підобластях апроксимації. Для відтворення білінійного Q -сплайна (2.30) розроблено структурну схему гібридного функціонального перетворювача двох змінних (ГФП-2).

При відтворенні апроксимуючої функції пристрій має виконувати такі операції:

1) збереження квантованих параметрів сплайна $q_{ij}^{[\alpha]}$ для кожної підобласті апроксимації $\Omega_{ij}=[x_i;x_{i+1}] \times [y_j;y_{j+1}]$;

2) вибірку слова, що складається з послідовності чотирьох кодів квантованих параметрів $\tilde{q}_{ij}^{[00]}$, $\tilde{q}_{ij}^{[10]}$, $\tilde{q}_{ij}^{[01]}$, $\tilde{q}_{ij}^{[11]}$, відповідно до значень незалежних змінних x , y , що надходять на вхід пристрою ;

3) обчислення по операціях виразу (2.29) за значеннями змінних x , y і обраним із блоку пам'яті відповідним параметрам сплайна.

Очевидно, що при будь-якій можливій структурі ГФП-2 у її складі необхідно мати адресний блок (АБ) і блок пам'яті (БП). Сьогодні промисловістю випускаються запам'ятовуючі пристрої, що конструктивно з'єднують в одному корпусі АБ і БП. У БП зберігатимемо цифрові коди $N_{ij}^{[\alpha]}$ параметрів $\tilde{q}_{ij}^{[\alpha]}$ функції, що апроксимує.

За основу побудови ГФП-2 взято спосіб одержання кожного доданка, а потім їхнього підсумовування. На рис. 2.19 наведено структурну схему пристрою.

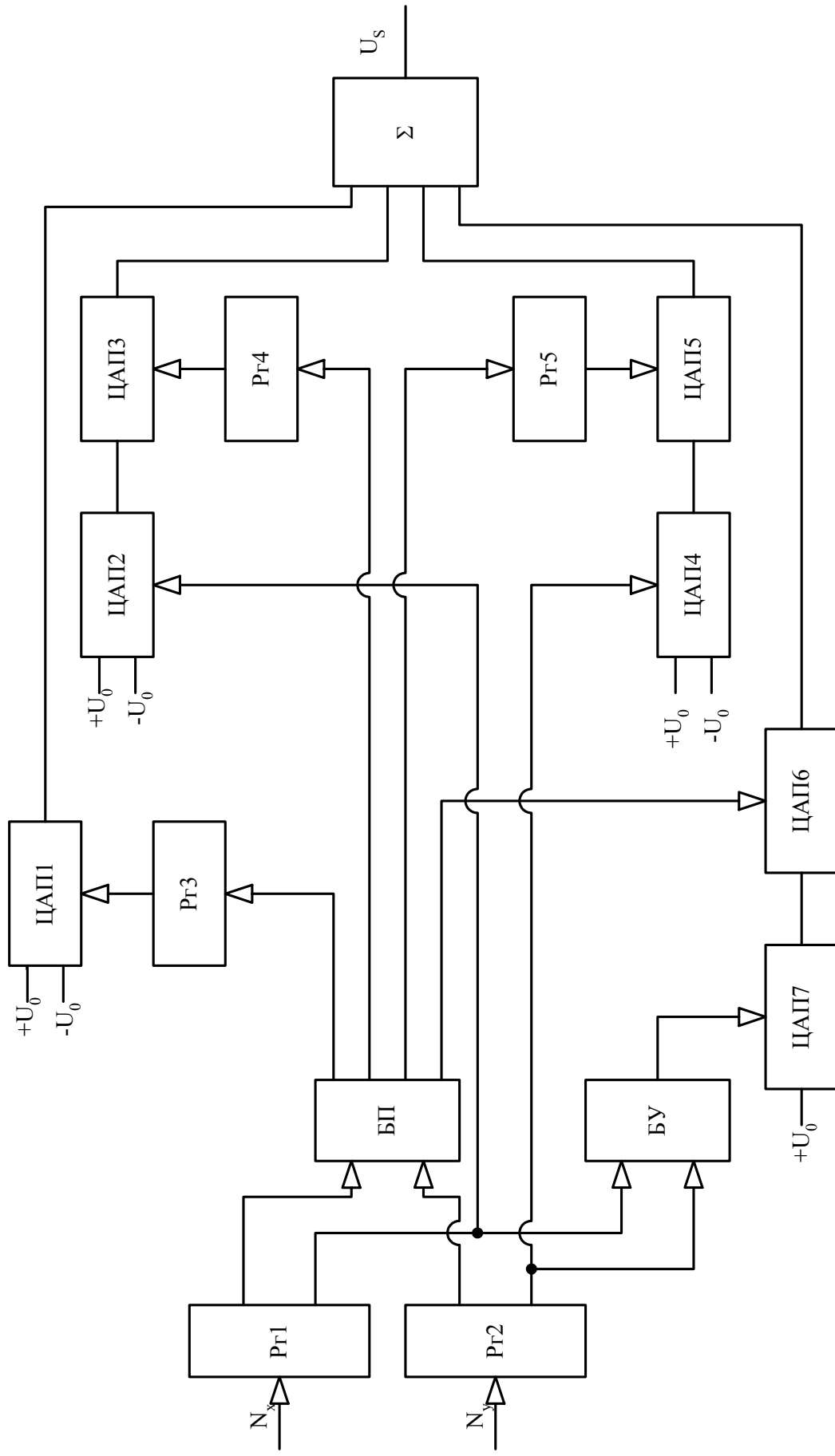


Рисунок 2.19 – Структурна схема ГФП для формування білінійного Q-сплайна на прямокутних підобластях апроксимації

Коди аргументів, що надходять по вхідних шинах, N_x і N_y записуються для збереження в регістри Rг1 і Rг2 відповідно. Старші розряди цих кодів надходять у БП і служать адресою для вибору параметрів сплайна, що відповідають заданій значеннями аргументів x і y підобласті апроксимації Ω_{ij} . Прочитана з БП інформація записується в регістри в такий спосіб: цифровий код $N_{ij}^{[00]}$ у Rг3, цифровий код $N_{ij}^{[10]}$ – у Rг2, цифровий код $N_{ij}^{[01]}$ – у Rг5, цифровий код $N_{ij}^{[11]}$ – у Rг6.

Для реалізації першого доданка $\tilde{q}_{ij}^{[00]}$ використовується ЦАПЗ, на цифровий вхід якого подається код $N_{ij}^{[00]}$ із Rг3, а на аналоговий вхід комутується в залежності від знакового розряду в Rг3 опорна напруга $+U_0$ чи $-U_0$. На виході ЦАПЗ з'являється напруга, що відповідає першому доданку.

Для реалізації другого доданка $\tilde{q}_{ij}^{[10]}(x - x_i)$ використовується ланцюжок, що складається з ЦАП1 і ЦАП2. На цифрові входи ЦАП1 надходять молодші розряди коду N_x , що являють собою приріст аргументу x : $(x - x_i)$. ЦАП1 перетворить цей код в аналогову величину: напругу. Потім ця напруга надходить на аналоговий вхід ЦАП2, на цифровий вхід якого надходить код $N_{ij}^{[10]}$ із Rг2. На виході ЦАП2 формується напруга, що відповідає добутку $\tilde{q}_{ij}^{[10]}(x - x_i)$.

Аналогічно реалізується третій доданок $\tilde{q}_{ij}^{[01]}(y - y_j)$, для чого використовуються молодші розряди коду N_y із Rг4, ЦАП4, що перетворить цифровий код приросту аргументу y в аналогову величину, і ЦАП5, що здійснює операцію множення приросту аргументу на постійний коефіцієнт $q_{ij}^{[01]}$.

Для відтворення четвертого доданка можна використовувати різні способи. Одним зі способів є застосування аналогового блоку множення (рис. 2.20). На цей блок з виходів ЦАП1 і ЦАП2 подаємо значення приростів аргументів в аналоговій формі, на виході блоку одержуємо добуток $(x - x_i)(y - y_j)$.

Для множення на постійний коефіцієнт $q_{ij}^{[11]}$, використовуємо ЦАП7, на цифровий вхід якого надходить код параметра $q_{ij}^{[11]}$, а на аналоговий – напруга, що відповідає добутку $(x-x_i)(y-y_j)$.

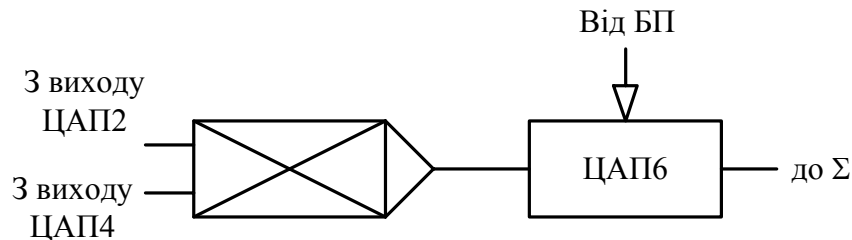


Рисунок 2.20 – Реалізація четвертого доданка з використанням аналогового блоку множення

Іншим способом є використання для операції множення тільки цифроаналогових перетворювачів (рис. 2.21). Так на вхід ЦАП6 подаємо напругу з виходу ЦАП4, що відповідає приросту аргументу $(y-y_j)$. На цифровий вхід ЦАП6 подаємо цифровий код, що відповідає приросту іншого аргументу $(x-x_i)$, а саме молодші розряди з $R_{г1}$. На виході формується напруга, що відповідає добутку $(x-x_i)(y-y_j)$. Для множення на $q_{ij}^{[11]}$ використовується, як і в попередньому способі ЦАП7.

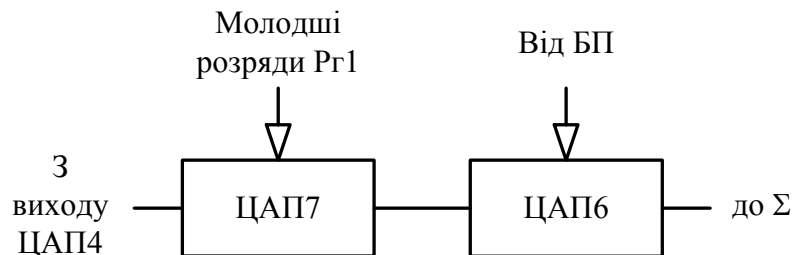


Рисунок 2.21 – Реалізація четвертого доданка з використанням цифроаналогових перетворювачів

Однак, ці способи погіршують точність і швидкодію ГФП-2, внаслідок зростання інструментальної похибки. Тому для запропонованої структури обраний третій спосіб, що не має цих недоліків.

Для відтворення четвертого доданка $q_{ij}^{[11]}(x-x_i)(y-y_j)$ у наведеній схемі використовується ланцюжок, який складається з цифрового блоку множення БУ, ЦАП6 і ЦАП7. БУ дозволяє отримати добуток $(x-x_i)(y-y_j)$ без похибки при подачі на вхід молодших розрядів кодів N_x і N_y , що відповідають приростам аргументів.

Такий блок може бути виконаний у вигляді матричного множника, або у вигляді блоку пам'яті, у якому адресними входами є входи для молодших розрядів кодів N_x і N_y , а на виході БУ формується цифровий код, що відповідає добутку $(x-x_i)(y-y_j)$. Потім за допомогою ЦАП7 проводиться множення коду добутку, перетвореного ЦАП6 у напругу з відповідним знаком, на код параметра $q_{ij}^{[11]}$.

Такий спосіб реалізації четвертого доданка характеризується досить високою швидкодією і високою точністю.

Потім усі чотири доданки надходять на входи аналогового суматора Σ . На його виході формується напруга, що відповідає значенню апроксимуючої функції при значеннях незалежних змінних x і y , що надійшли на вхід.

Перед початком роботи ГФП-2 проводиться його програмування на основі попередньо розрахованих таблиць апроксимації. Усі регістри обнуляються. У процесі програмування пристрою в БП мають бути занесені кодові значення параметрів сплайна з їх знаками: $N_{ij}^{[\alpha]}$.

Для розрахунку параметрів $N_{ij}^{[\alpha]}$ і $\tilde{q}_{ij}^{[\alpha]}$, використовуваних для відтворення Q -сплайна пристроєм запропонованої структури, використовується алгоритм, описаний у підрозділі 2.2. Машинна реалізація описаного алгоритму виконана у вигляді програми *SPLDL*.

Структура програми включає: основну програму, підпрограму *KB4* – для

квантування коефіцієнтів сплайна з мінімізацією за кількістю рівнів квантування.

Для роботи програми потрібна така інформація:

- границі області визначення функції – значення a_1, a_2, b_1, b_2 ;
- припустима відносна приведена похибка апроксимації ε у відсотках;
- кількість перевірних точок $M1, M2$.

У залежності від кількості перевірних точок визначаються розмірності масивів: $SA(M1 \times M2)$ – масив коефіцієнтів сплайна; $KA(M1 \times M2)$ – масив кодів коефіцієнтів сплайна.

Вихідні дані програми містять: таблицю квантованих параметрів сплайна і їхніх цифрових кодів; фактичну похибку апроксимації у відсотках; кількість розподілів на підобласті апроксимації K_1 і K_2 (у програмі відповідно $LO(1)$, $LO(2)$); значення Δ , Δ_n^* , $\Delta_{кв}^*$ (у програмі позначені EM, EN, EK). Якщо похибка апроксимації перевищує припустиму величину і за заданих умов немає можливості її зменшити, то з'являється повідомлення: "Точність не досягнута".

Реалізація сплайна на трикутних підобластях апроксимації. Під час аналізу виразів (2.42) і (2.43) видно, що вигляд сплайна однаковий для обох трикутників, розрізняються лише значення параметрів B_{ij}^T і C_{ij}^T для кожного трикутника. Отже, структура перетворювача для відтворення Q -сплайна на трикутних підобластях апроксимації може залишатися такою ж, як і для ФП-2. При цьому необхідно апаратно реалізувати процедуру визначення трикутника, у якому знаходиться точка, задана парою значень аргументів $(\bar{x}_i; \bar{y}_j)$ і вибірку відповідних параметрів.

Ці задачі були розв'язані і втілені в структурі ФП-2, що відтворює лінійний Q -сплайн на трикутних підобластях апроксимації. Структурна схема пристрою зображена на рис. 2.22.

У пристрої незалежна змінна x моделюється двійковим кодом N_x , отриманим після перетворення вхідної напруги U_x на АЦП1, а незалежна змінна y – двійковим кодом N_y , отриманим на виході АЦП2. Оскільки вся область

визначення Ω функції $f(x,y)$ розбивається на конгруентні прямокутні підобласті, загальна кількість яких дорівнює цілому степеню двійки, то прирости аргументів $\Delta x = x - x_i$, $\Delta y = y - y_j$ моделюються молодшими розрядами кодів N_x і N_y , відповідно, які позначимо $N_{\Delta x}$ і $N_{\Delta y}$. Подамо незалежні змінні у вигляді:

$$x = N_x \cdot \delta x,$$

$$y = N_y \cdot \delta y,$$

де δx і δy – кроки квантування незалежних змінних.

З урахуванням цього перепишемо вираз (2.38) у вигляді:

$$(N_{x_{i+1}} \delta x - N_{x_i} \delta x)(N_{y_j} \delta y - N_{y_{j+1}} \delta y) - (N_{y_{j+1}} \delta y - N_{y_j} \delta y)(N_x \delta x - N_{x_i} \delta x) = 0,$$

або після ділення на добуток $(\delta x \cdot \delta y)$

$$(N_{x_{i+1}} - N_{x_i})(N_{y_j} - N_{y_{j+1}}) - (N_{y_{j+1}} - N_{y_j})(N_x - N_{x_i}) = 0.$$

Співмножники $(N_x - N_{x_i})$ і $(N_y - N_{y_j})$ є кодом приросту $N_{\Delta x}$ й $N_{\Delta y}$ аргументів. Кожний зі співмножників $(N_{x_{i+1}} - N_{x_i})$ і $(N_{y_{j+1}} - N_{y_j})$ є постійним двійковим числом, що виражається одиницею з нулями. Останній вираз подамо у вигляді:

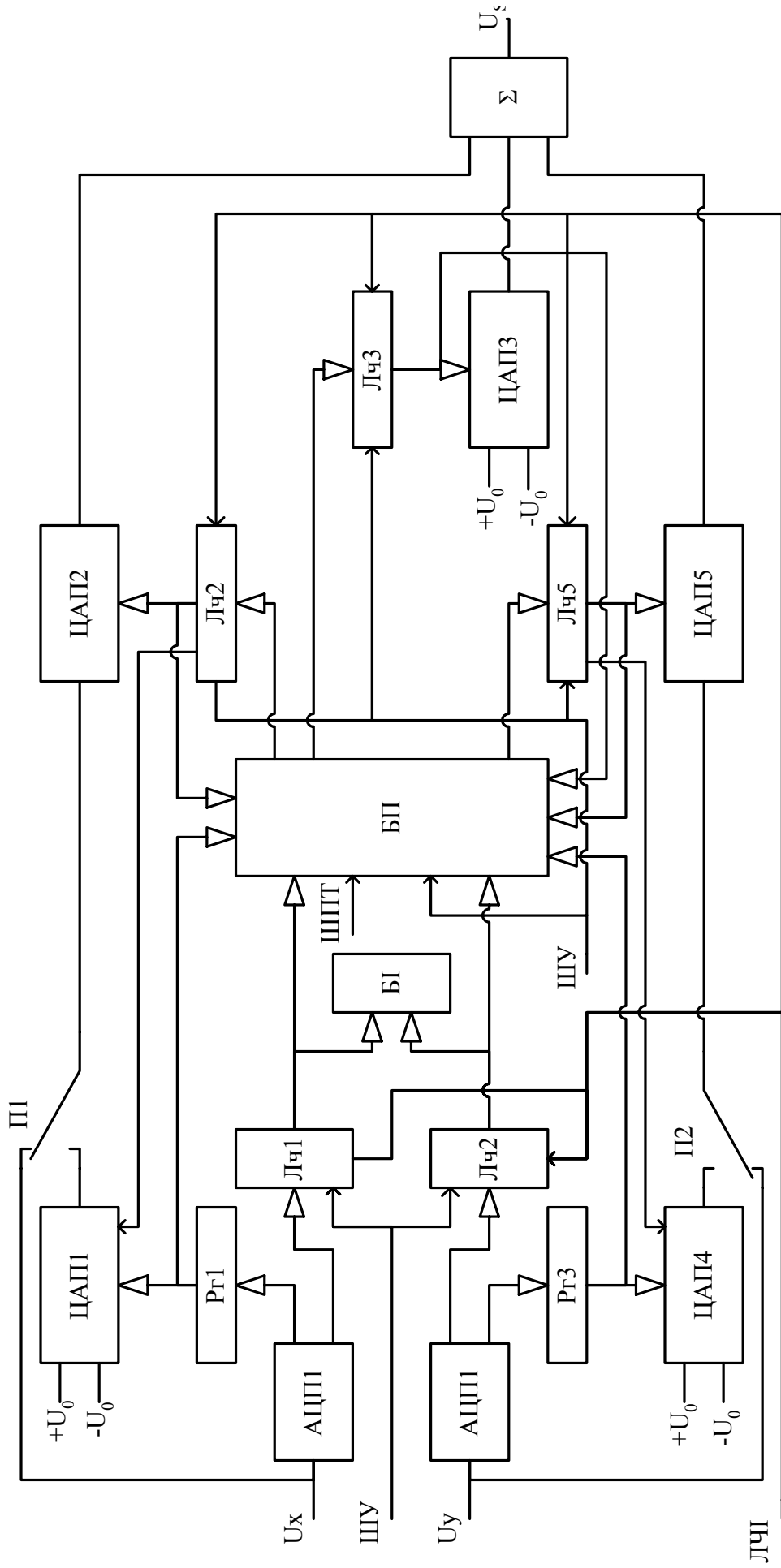


Рисунок 2.22 – Структурна схема ФП-2 на трикутних підобластях апроксимації

$$(N_{x_{i+1}} - N_{x_i}) \cdot N_{\Delta y} - (N_{y_{j+1}} - N_{y_j}) \cdot N_{\Delta x} = 0. \quad (2.74)$$

Якщо кількість розрядів у кодї $N_{\Delta y}$ більше, ніж у кодї $N_{\Delta x}$, то вираз (2.74) подамо у вигляді:

$$N_{\Delta y} = \left(\frac{N_{y_{j+1}} - N_{y_j}}{N_{x_{i+1}} - N_{x_i}} \right) \cdot N_{\Delta x}. \quad (2.75)$$

Якщо ж кількість розрядів більше в кодї $N_{\Delta x}$, то вираз (2.74) подамо у вигляді:

$$N_{\Delta y} \cdot \left(\frac{N_{x_{i+1}} - N_{x_i}}{N_{y_{j+1}} - N_{y_j}} \right) = N_{\Delta x}. \quad (2.76)$$

Вирази, що стоять у дужках, у співвідношеннях (2.75), (2.76) є постійними числами, що виражаються одиницею з нулями. А помножити цифровий код на одиницю з нулями означає зрушення цього коду убік старших розрядів. Ця процедура буде виконана автоматично під час настроювання перетворювача на апроксимацію заданої функції. При цьому розряди кодів $N_{\Delta x}$ і $N_{\Delta y}$ з виходів АЦП1 і АЦП2 будуть підключені до старших розрядів регістрів Rг1 і Rг2, а незадіяні розряди заповнюються нулями.

Залишається лише порівняти коди. Для цієї мети служить схема порівняння кодів (СПК), зображена на рис. 2.23. Результатом порівняння буде ознака трикутника $T(x,y)$, що дорівнює “0” при влученні точки (x,y) у «перший» трикутник, чи дорівнює “1” при влученні точки в «другий» трикутник (у першому випадку ліві частини виразів (2.75) і (2.76) більше правих частин, у другому випадку – менше). Ця ознака знімається з виходу СПК і подається через логічний елемент АБО на молодший розряд адресного входу ЗП2, з якого вибираються необхідні коди: або параметрів $B_{ij}^{[0]}$ і $C_{ij}^{[0]}$, або $B_{ij}^{[1]}$ і $C_{ij}^{[1]}$. Для даного способу триангуляції і реалізованих рівнянь апроксимуючих поверхонь вигляду (2.42) і (2.43) параметр $A_{ij}=f(x_i;y_j)$ – загальний для обох трикутників, тому його збереження проводиться в окремому ЗП1.

Отже, при введенні незначних апаратурних витрат (схеми порівняння кодів, логічного елемента АБО, розбивки запам'ятовуючого пристрою на два блоки), отримуємо більш точний спосіб апроксимації функції лінійним сплайном.

Пропонований ФП відноситься до класу універсальних перетворювачів, тому що дозволяє відтворювати різні функціональні залежності. Для переходу від відтворення однієї функції до іншої призначений режим програмування.

У цьому режимі на шині керування ШУ діє логічний "0". При цьому блок пам'яті БП переводиться в режим запису кодових значень параметрів апроксимації, а на всі лічильники подається дозвіл на підрахунок імпульсів, що ідуть по шині лічильних імпульсів ЛЧІ. Одночасно перемикачі П1 і П2 встановлюються в таке положення, при якому аналогові входи ЦАП2 і ЦАП5 підключаються, відповідно, до шин введення вхідних напруг U_x і U_y , що моделюють аргументи. Передбачається також, що попередньо виконано стирання інформації в пам'яті і всі лічильники обнулені.

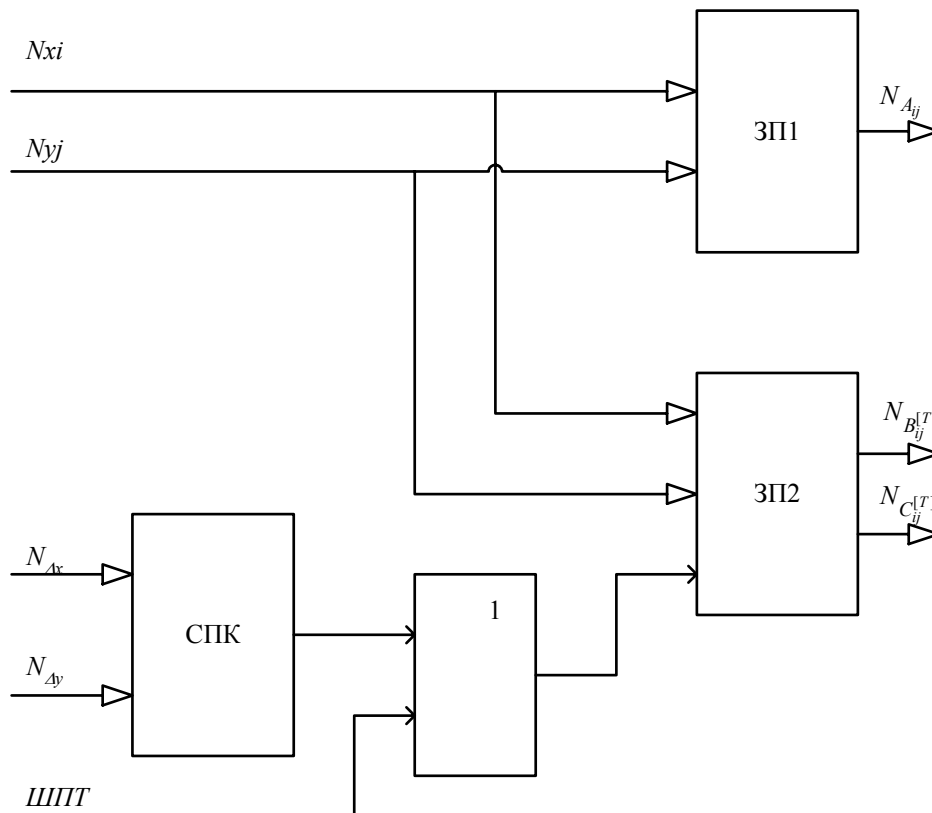


Рисунок 2.23 – Структурна схема блоку пам'яті

Підставою для настроювання функціонального перетворювача є множина:

$$\{f(x_i, y_j)\}, \quad x_i = a + ih_x, \quad y_j = b + jh_y, \quad i = \overline{0, K_1}, \quad j = \overline{0, K_2}.$$

Мета програмування в тому, щоб для всіх i і j записати в блок пам'яті кодові значення параметрів апроксимації.

Розглянемо процедуру програмування для довільних i і j у вигляді послідовності наступних шести кроків.

1. На шині введення аргументів подаються нульові напруги $U_x=0$ і $U_y=0$. На лічильники ЛЧ1 і ЛЧ4 послідовно подають лічильні імпульси по шині ЛЧ1 доти, доки блок індикації БІ зафіксує необхідний номер підобласті апроксимації, що визначається сукупністю кодових значень лічильників ЛЧ1 і ЛЧ2. Після цього організується подача лічильних імпульсів по шині ЛЧ1 на лічильник ЛЧ3 до накопичення на ньому такого коду, при якому напруга на виході суматора Σ , обумовлена вихідною напругою ЦАП3, стане відповідати $f(x_i, y_j)$ з необхідним знаком. У лічильнику ЛЧ3 буде сформований код початкової аплікати (2.43).

У наступних двох кроках роблять програмування для «першого» трикутника. При цьому на шину ознаки трикутника ШПТ подають логічний «0» і, тому що на СПК подають коди N_{Ax} і N_{Ay} рівні «0», то на виході елемента АБО буде «0», що кодує влучення точки (x, y) у «перший» трикутник.

2. На шині введення другого аргументу залишають нульову напругу $U_y=0$, у той час як на шину введення першого аргументу подають напругу, що відповідає приросту аргументу h_x . Потім по шині ЛЧ1 направляють лічильні імпульси на лічильник ЛЧ2 доти, доки напруга на виході ЦАП2, при сумуванні з раніше сформованою напругою на виході ЦАП3 стане відповідати $f(x_i + h_x; y_j) = f(x_{i+1}; y_j)$. Тепер у лічильнику ЛЧ2 сформований код, що відповідає (2.46).

3. На шині введення першого аргументу залишають напругу, що відповідає h_x , водночас на шину введення другого аргументу подають напругу,

що відповідає h_y . Після цього організується подача лічильних імпульсів із шини ЛЧІ на лічильник ЛЧ5 до накопичення на ньому такого коду, при якому напруга на виході суматора Σ стане відповідати $f(x_i + h_x; y_j + h_y) = f(x_{i+1}; y_{j+1})$. При цьому в лічильнику буде сформований код, що відповідає $C_{ij}^{[0]}$ (2.47).

Після виконання цього кроку програмування з лічильника ЛЧ3 переписується код у ЗП1 за адресою, зазначеною кодами в лічильниках ЛЧ1 і ЛЧ4, а коди з лічильників ЛЧ2 і ЛЧ5 – у ЗП2 в одну комірку за адресою, зазначеною кодами в лічильниках ЛЧ1 і ЛЧ4 і бітом інформації, що знімається з виходу елемента АБО.

2. Тепер проводять програмування для «другого» трикутника. Для цього на шини введення аргументів подають нульові напруги $U_x = U_y = 0$. На шину ШПТ подають логічну “1”, що з виходу елемента АБО кодуватиме влучення точки (x, y) у «другий» трикутник. Обнуляють лічильники СЧ2 і СЧ5. При цьому на виході суматора Σ має установитися напруга, що відповідає $f(x_i; y_j)$.

5. На шині введення першого аргументу залишають нульову напругу $U_x = 0$, у той час як на шину введення другого аргументу, подають напругу, що відповідає h_y . Потім по шині ЛЧІ в лічильник ЛЧ5 направляють лічильні імпульси доти, доки напруга з виходу ЦАП5, при сумуванні з раніше сформованою напругою з виходу ЦАП3, стане відповідати $f(x_i; y_j + h_y) = f(x_i; y_{j+1})$. Тепер у лічильнику ЛЧ5 буде сформований код, що відповідає $C_{ij}^{[1]}$ (2.48).

6. На шині введення другого аргументу залишають напругу, що відповідає h_y . На шину введення першого аргументу подають напругу, що відповідає h_x . Після цього організується подача лічильних імпульсів по шині ЛЧІ на лічильник ЛЧ2 до накопичення в ньому такого коду, при якому напруга на виході суматора Σ стане відповідати $f(x_i + h_x; y_j + h_y) = f(x_{i+1}; y_{j+1})$. При

цьому в лічильнику ЛЧ2 буде сформований код, що відповідає $B_{ij}^{[1]}$ (2.49).

Після виконання шостого кроку програмування коди з лічильників ЛЧ2 і ЛЧ5 переписуються в ЗП2 за адресою, зазначеною кодами в лічильниках ЛЧ1 і ЛЧ4 та бітом інформації на виході логічного елемента АБО.

Аналогічно виконується програмування для всіх інших підобластей апроксимації.

У робочому режимі на шині керування ШУ діє логічна “1”, що переводить усі лічильники в режим прийому-передачі паралельних кодів, що діють на їхніх настановних входах. Одночасно цей же сигнал переводить блок пам'яті БП із режиму запису в режим зчитування кодів. У робочому режимі перемикачі П1 і П2 мають приєднувати відповідно вихід ЦАП1 до аналогового входу ЦАП2 і вихід ЦАП4 до аналогового входу ЦАП5. На шину ШПТ подають логічний “0”.

Напруги U_x і U_y , що надійшли на вхідні шини і моделюють незалежні змінні, за допомогою АЦП1 і АЦП2 перетворюються в пропорційні двійкові коди N_x і N_y . Старші розряди цих кодів $N_{\Delta x}$ і $N_{\Delta y}$ надходять відповідно в лічильники ЛЧ1 і ЛЧ2, що є адресними регістрами блоку пам'яті. Одночасно блок індикації БІ вказує номер підобласті, у якій знаходиться точка (x,y) , що задається вхідними напругами. Молодші розряди кодів аргументів $N_{\Delta x}$ і $N_{\Delta y}$, що моделюють прирости аргументів, надходять для оперативного збереження в регістри Рг1 і Рг2. З цих регістрів коди $N_{\Delta x}$ і $N_{\Delta y}$ надходять на схему порівняння кодів СПК, що виробляє сигнал приналежності точки (x,y) до того чи іншого трикутника – ознаку трикутника $T(x,y)$, що через логічний елемент АБО надходить на молодший розряд дешифратора адреси ЗП2. Отже, на дешифратор адреси ЗУ1 надходить код, що складається з кодів N_{xi} і N_{yj} , за яким вибирається код параметра A_{ij} , і записується в лічильник ЛЧ3. На дешифратор адреси ЗП2 надходять коди N_{xi} і N_{yj} і біт інформації з виходу СПК. За цією адресою з ЗП2 вибирається пара кодів: чи $\tilde{B}_{ij}^{[0]}$, $\tilde{C}_{ij}^{[0]}$ якщо на виході СПК – логічний “0”, або

$\tilde{B}_{ij}^{[1]}$, $\tilde{C}_{ij}^{[1]}$, якщо на виході СПК – логічна “1”. Код параметра $\tilde{B}_{ij}^{[0]}(\tilde{B}_{ij}^{[1]})$ записується в лічильник ЛЧ2, а код параметра $\tilde{C}_{ij}^{[0]}(\tilde{C}_{ij}^{[1]})$ записується в лічильник ЛЧ5. Старший розряд коду параметра $\tilde{B}_{ij}^{[0]}(\tilde{B}_{ij}^{[1]})$ будучи знаковим, керує полярністю опорної напруги U , що підключається до аналогового входу ЦАП1. Старший розряд коду параметра $\tilde{C}_{ij}^{[0]}(\tilde{C}_{ij}^{[1]})$, знаковий розряд, керує полярністю опорної напруги U_0 , що підключається до аналогового входу ЦАП2. На виході ЦАП1, на цифрові входи якого надходить код N_{dx} із Рг1, формується напруга, що відповідає добутку $\text{sgn}(\tilde{B}_{ij}^{[T]})(x - x_i)$, що надходить на аналоговий вхід ЦАП2. На цифрові входи ЦАП2 надходить код, що відповідає модулю параметра $\tilde{B}_{ij}^{[T]}$. На ЦАП2 відбувається перемноження величин $\text{sgn}(\tilde{B}_{ij}^{[T]}) \times (x - x_i)$ і $|\tilde{B}_{ij}^{[T]}|$. Отже, на виході ЦАП2 формується другий доданок у виразі (2.42) чи (2.43).

Аналогічно за допомогою елементів Рг2, ЦАП4, ЛЧ5 і ЦАП5 формується третій доданок виразу (2.42) чи (2.43).

Перший доданок формується на виході ЦАП3, як перетворення цифрового коду із ЛЧ3 у напругу.

Напруги з виходів ЦАП2, ЦАП3, ЦАП5, що моделюють кожен доданок у виразі (2.42) надходять на входи суматора Σ , на виході якого формується напруга, що відповідає значенню апроксимуючого лінійного Q -сплайна в точці (x,y) .

У такий же спосіб обчислюється значення апроксимуючого лінійного Q -сплайна в будь-якій точці (x,y) , обумовленої вхідними напругами U_x і U_y .

Роботі пристрою не обов'язково має передувати етап програмування в тому вигляді, у якому він описаний вище. Усі, параметри апроксимації можна розрахувати заздалегідь на ЕОМ і записати в блок пам'яті коди цих параметрів. Для такого розрахунку розроблена програма *TREUG1*.

Структура програми включає: основну програму, підпрограму *PROV* – для перевірки виконання співвідношення (2.44), підпрограму *КВАНТ* – квантування параметрів апроксимації, що обчислюються, (алгоритм квантування описаний у підрозділі 2.1).

Для роботи програми потрібна така інформація:

- межі області визначення функції $f(x,y)$ – значення a_1, a_2, b_1, b_2 ;
- припустима відносна приведена похибка апроксимації ε у відсотках;
- розрядності цифрових кодів для подання незалежних змінних m_1 і m_2 .

Залежно від вихідних даних, що вводяться, визначається розрядність масивів $SA0(2^{m_1} \cdot 2^{m_2})$ – масив точних коефіцієнтів сплайна; $SA(2^{m_1} \cdot 2^{m_2})$ – масив квантованих коефіцієнтів сплайна; $KA(2^{m_1} \cdot 2^{m_2})$ – масив цифрових кодів коефіцієнтів сплайна.

Вихідні дані програми містять: таблицю квантованих параметрів сплайна і їхніх цифрових кодів; максимальну фактичну абсолютну і відносну приведену (у відсотках.) похибку апроксимації Q -сплайном; розрядності кодів N_{dx} і N_{dy} (у програмі позначена $KRDX, KRDY$); кількість підобластей апроксимації по осі x і по осі y (позначені KOX і KOY); розміри підобластей апроксимації по осі x і по осі y (позначена H, θ); розрядність кодів для подання параметрів. Якщо похибка апроксимації сплайном з точним поданням параметрів перевищує припустиму величину, то з'являється повідомлення: «Точність не досягнута». Якщо похибка апроксимації лінійним Q -сплайном перевищує припустиму величину і збільшенням розрядності коефіцієнтів сплайна до 16 немає можливості зменшити її до заданої, з'являється повідомлення: «При розрядності коефіцієнтів 16 точність не досягнута». Слід зважати на те, що в цьому випадку сплайн із точними параметрами забезпечує задану точність апроксимації.

Програма *TREUGI* успішно випробувана для ряду функцій. Результати розрахунку для деяких функцій наведені в таблиці 2.3.

Таблиця 2.3 – Результати розрахунку програми *TREUGI*

Вид функції, область визначення; $f(x,y)$, $\Omega=[a_1;a_2] \times [b_1;b_2]$	Кількість підобластей апроксимації	Розрядність параметрів	Відносна приведена похибка,%
$x \cdot y$ [0;1]×[0;1]	16×2=32	8	0,8631
	8×4=32	10	0,8377
	4×8=32	12	0,8094
	2×16=32	14	0,8082
		16	0,8065
		∞	0,8063
$x \cdot y$ [0;1]×[0;1]	16×4=64	8	0,5264
	8×8=64	10	0,4339
	4×16=64	12	0,4047
		14	0,4053
		16	0,4036
		∞	0,4031
$x \cdot y$ [0;1]×[0;1]	16×8=128 8×16=128	8	0,3565
		10	0,2382
		12	0,2045
		14	0,2041
		16	0,2022
		∞	0,2016
	16×16=256	8	0,2786
		10	0,1433
		12	0,1102
		14	0,1036
		16	0,1015
		∞	0,1008
$\ln(x \cdot y)$; [0,5;5]× [0,5;5]	16×16=256	8	1,6133
		10	1,5820
		12	1,5741
		14	1,5722
		16	1,5732
		∞	1,5733

Вид функції, область визначення; $f(x,y),$ $\Omega=[a_1;a_2] \times [b_1;b_2]$	Кількість підобластей апроксимації	Розрядність параметрів	Відносна приведена похибка, %
$\sqrt{x^2 + y^2}$ [0;10]×[0;10]	8×8=64	8	0,8905
		10	0,7715
		12	0,8070
		14	0,8001
		16	0,7997
		∞	0,7998
$\sqrt{x^2 + y^2}$ [0;10]×[0;10]	16×16=256	8	0,3594
		10	0,3795
		12	0,3989
		14	0,3981
		16	0,3994
		∞	0,3999
$x^2 + y^2$ [0;3,5]× [0;3,5]	16×4=64 4×16=64	8	0,9962
		10	0,8947
		12	0,8654
		14	0,8585
		16	0,8572
		∞	0,8566
	8×8=64	8	0,5581
		10	0,4413
		12	0,4128
		14	0,4056
		16	0,4037
		∞	0,4C31
$x^2 + y^2$ [0;3,5]× [0;3,5]	16×8=128	8	0,4172
		10	0,2939
		12	0,2626
		14	0,2545
		16	0,2526
		∞	0,2520
	16×16=256	8	0,2759

Вид функції, область визначення; $f(x,y),$ $\Omega=[a_1;a_2] \times [b_1;b_2]$	Кількість підобластей апроксимації	Розрядність параметрів	Відносна приведена похибка, %
$x^2 + y^2$ [0;3,5]× [0;3,5]	16×16=256	10	0,1451
		12	0,1122
		14	0,1035
		16	0,1015
		∞	0,1008

У таблиці знаком ∞ відзначено рядки, у яких наведено інформацію для сплайна з точним поданням параметрів

Реалізація сплайна на трикутних підобластях апроксимації зі змішаними видами триангуляції. Для технічної реалізації наведеного в підрозділі 2.3.2 алгоритму розроблено структуру пристрою, що наведено на рис.2.22. Ця структура подібна наведеній в підрозділі 2.5.2.

Для перетворення вхідних аналогових сигналів U_x і U_y , що моделюють значення вхідних незалежних змінних, у двійковій коді застосовуються аналого-цифрові перетворювачі АЦП1 та АЦП2, відповідно. Отримані коди N_x і N_y , якими аргументи моделюються в пристрої, записуються в регістри Rг1 і Rг2 для оперативного збереження.

Розглянутий пристрій має реалізувати вираз (2.70), що перетворимо у такий вигляд:

$$\begin{aligned}
 S_{ij}^{RT}(x,y) = & \operatorname{sgn}\left(A_{ij}^{RT}\right) \cdot N_{A_{ij}^{RT}} \cdot \delta A + \\
 & + \operatorname{sgn}\left(B_{ij}^{RT}\right) \cdot N_{B_{ij}^{RT}} \cdot \delta B(N_x - N_{x_i} - R_{ij}N_{h_x}) \cdot \delta x + \\
 & + \operatorname{sgn}\left(C_{ij}^{RT}\right) \cdot N_{C_{ij}^{RT}} \cdot \delta C(N_y - N_{y_j}) \cdot \delta y,
 \end{aligned} \tag{2.77}$$

де $\operatorname{sgn}(P)$ – знак параметра P , N_P – код параметра P , ΔP – крок квантування параметра P за рівнем

$$\delta P = \frac{|P|_{\max}}{2^l - 1},$$

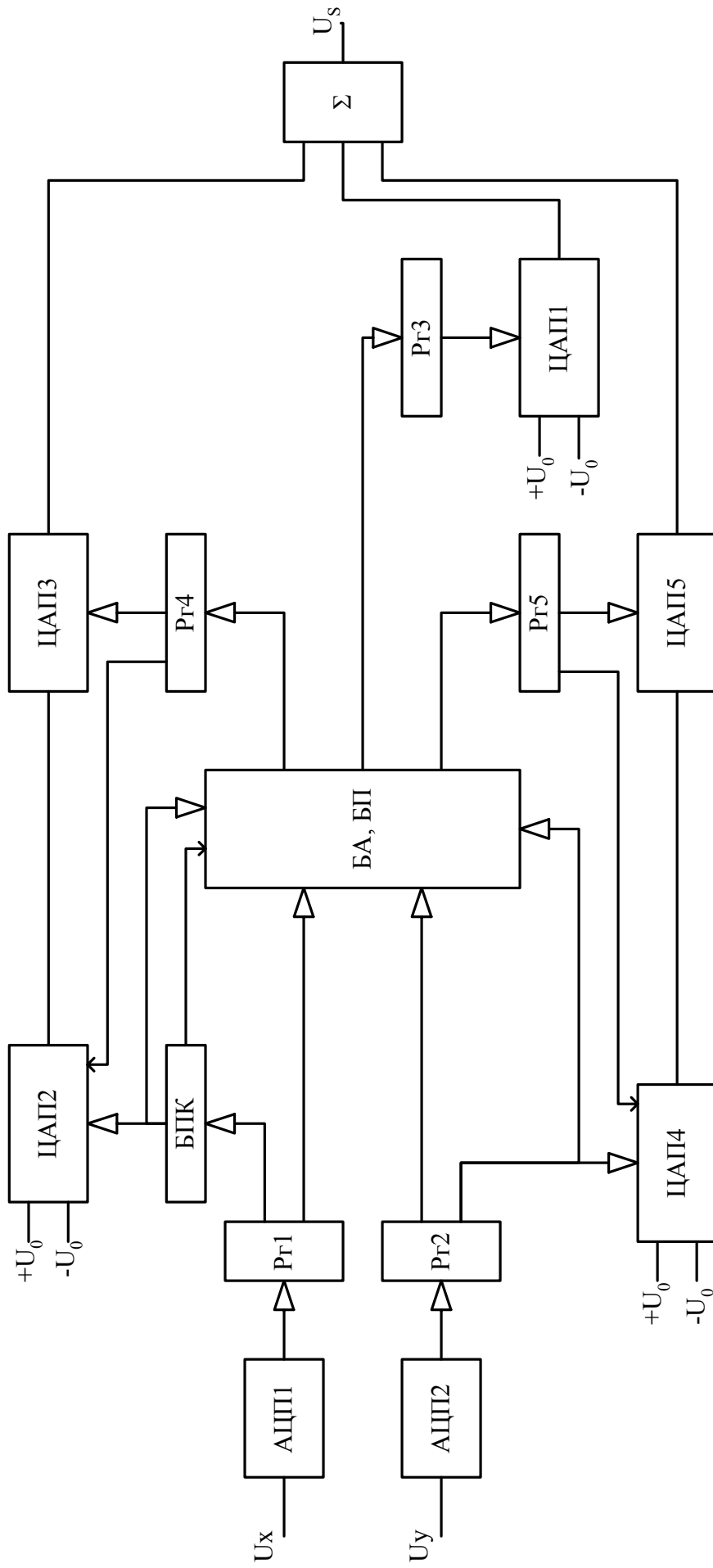


Рисунок 2.24 – Структурна схема ФП-2 на трикутних підобластях апроксимації з різними видами триангуляції

де $|P|_{\max} = \max_{\forall ij} \{|P_{ij}|\}$, l – розрядність слова для подання параметра P , R_{ij} – ознака розбивки прямокутної підобласті Ω_{ij} на трикутники ($R_{ij}=0$ – «пряма», $R_{ij}=1$ – «зворотна» розбивка).

Розглянемо роботу пристрою.

Старші розряди кодів аргументів, що позначимо N_{xi} і N_{yj} , які визначають прямокутну підобласть Ω_{ij} , подаються на блок адресації (БА) і блок пам'яті (БП). За цими кодами із БП із першого запам'ятовуючого пристрою (ЗП1) вибирається ознака розбивки підобласті Ω_{ij} , на трикутники – R_{ij} , що направляється на блок передачі коду (БПК). БПК використовується для того, щоб сформувавши код, описуваний у (2.77) співмножником $(N_x - N_{xi} - R_{ij}N_{hx})$. Для $R_{ij}=0$ вираз має вигляд:

$$(N_x - N_{x_i}) = N_{\Delta x},$$

тобто цей код являє собою молодші розряди коду N_x – код збільшення аргументу x . У цьому випадку БПК має передавати молодші розряди коду N_x без зміни. Для $R_{ij}=1$ маємо:

$$N_x - N_{x_i} - N_{h_x} = N_x - N_{x_{i+1}} = -N_{\Delta x},$$

тобто цей код являє собою додатковий код коду $N_{\Delta x}$, із протилежним знаком. Отже, при $R_{ij}=1$ БПК має формувати додатковий код, а саме:

$$N_{\Delta x} = \bar{N}_{\Delta x} + 1.$$

Знак «-» змінюватиме знак параметра B_{ij}^{RT} , на протилежний. Отже, пристрій реалізуватиме такий вираз для другого доданка:

$$(-1)^{R_{ij}} \cdot \text{sgn}(B_{ij}^{RT}) \cdot N_{B_{ij}^{RT}} \cdot \delta B \cdot |N_x - N_{x_i} - R_{ij}N_{h_x}| \cdot \delta x. \quad (2.78)$$

Схема БПК наведена на рис.2.15. На виході БПК з'являтиметься код $N_{\Delta x}$ для «прямої» розбивки ($R_{ij}=0$), або $N_{\Delta x}$ для «зворотної» розбивки. Позначимо ці коди одним символом $N_{\Delta x}$. За кодами N_{xi} , N_{yj} , $N_{\Delta x}$ і $N_{\Delta y}$ блок адресації (БА) формує адреси, що надходять на блок пам'яті.

Розглянемо спосіб збереження параметрів і структуру блоку адресації та блоку пам'яті. Для кожної прямокутної підобласті Ω_{ij} , (тобто в осередках, обумовлених адресою, складених з кодів N_{xi} і N_{yi}) зберігаються три параметри: A_{ij}^0 , B_{ij}^{00} і C_{ij}^{01} . Всі інші параметри вибираються із сусідніх комірок, як це показано в підрозділі 2.3.2. Крім того, в окремих ЗП необхідно зберігати параметри правої і верхньої межі області Ω , а саме: $A_{K_1j}^1$ ($j=\overline{1, K_2}$), $B_{iK_2}^{01}$ ($i=\overline{1, K_1}$), $C_{K_2j}^{00}$ ($j=\overline{1, K_2}$). Це зроблено для того, щоб не дуже ускладнювати блок адресації.

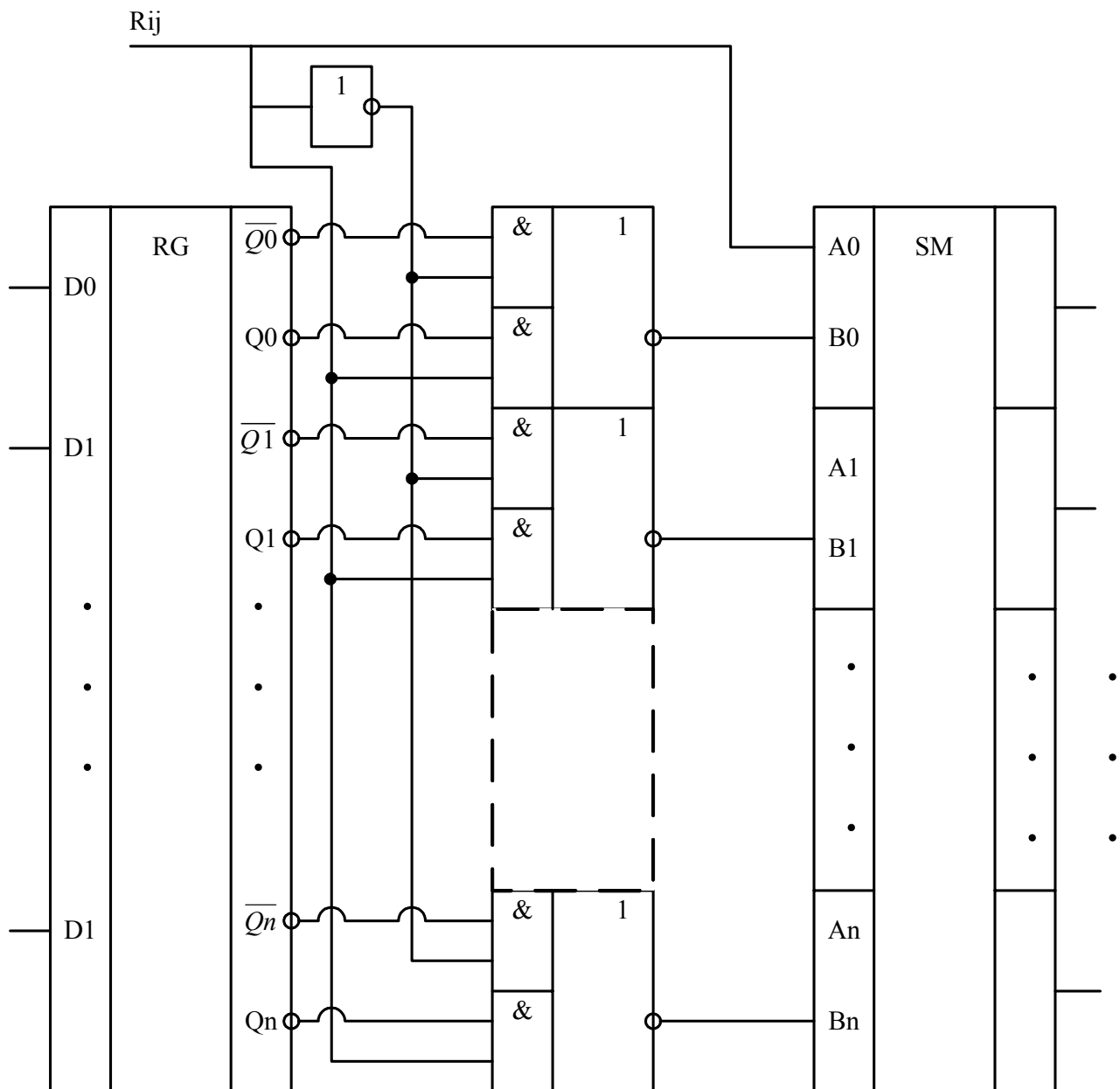


Рисунок 2.25 – Схема блоку передачі коду
Структурну схему БА і БП наведено на рис. 2.26.

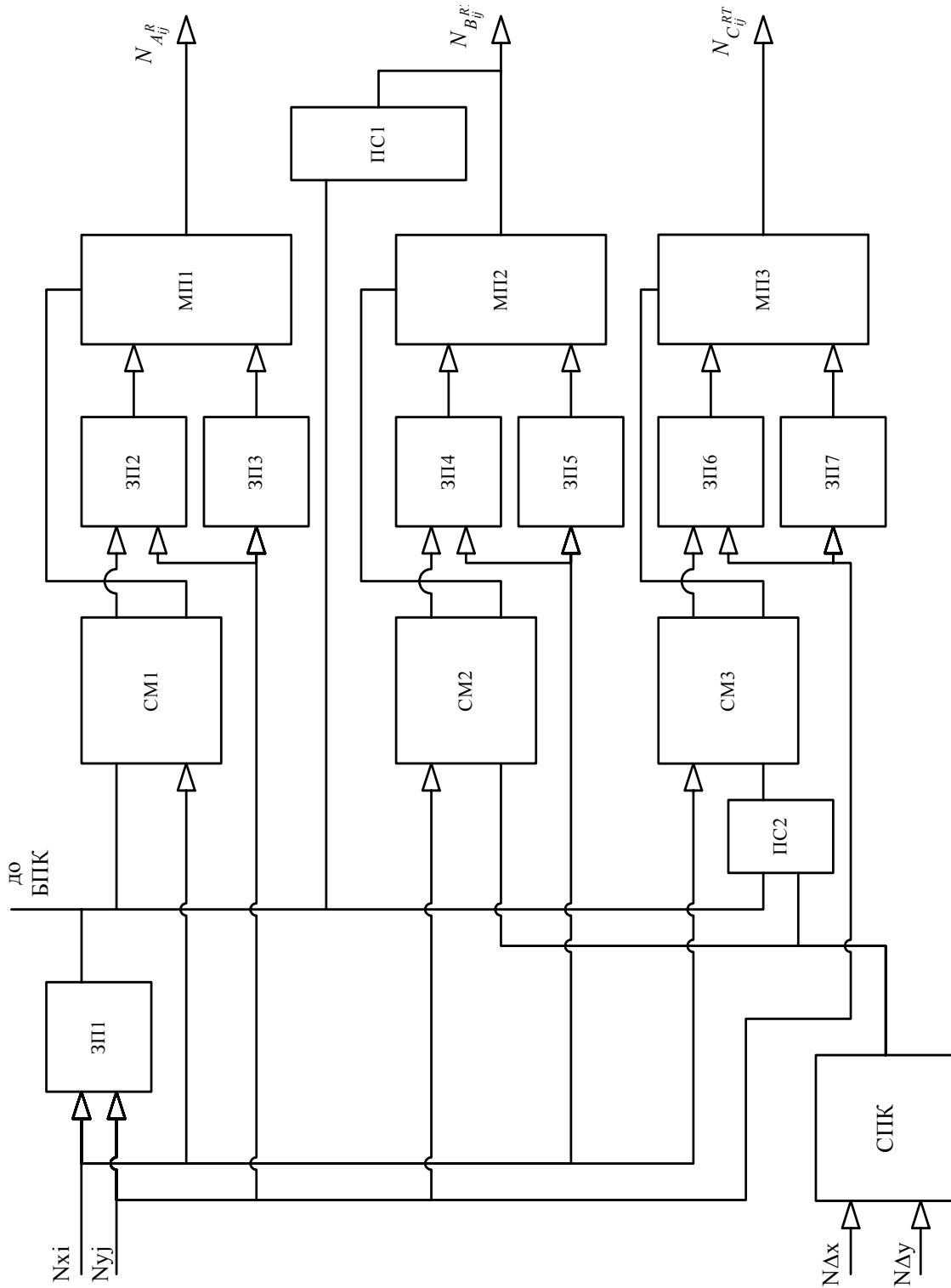


Рисунок 2.26 – Структурна схема блока адресації і блока пам'яті

Коди N_{xi} і N_{yi} , що визначають адресу прямокутної підобласті апроксимації, надходять на ЗП1, у якому за кожною адресою зберігається тільки один біт інформації R_{ij} , що вказує вид триангуляції підобласті Ω_{ij} . Ця ознака розбивки надходить із БП на БПК.

На схему порівняння кодів (СПК) надходять коди збільшення аргументів $N_{\Delta x}$ і $N_{\Delta y}$. СПК відповідно до виразу (2.69) виробляє ознаку трикутника $T(x,y)$: «0» – для «першого» трикутника, «1» – для «другого» трикутника.

Для формування адреси осередку, у якому знаходиться код параметра A_{ij}^R і використовується суматор ($\Sigma 1$). На вхід суматора подається код N_{xi} і R_{ij} . Сума цих кодів визначає молодші розряди адресного коду, старші розряди коду визначає код $N_{\Delta y}$, що безпосередньо подається на адресний вхід ЗП2. Якщо $R_{ij}=0$, тобто розбивка «пряма», то необхідно вибирати параметр A_{ij}^0 з комірки з адресою, складеною із кодів N_{xi} , N_{yj} . При цьому на керуючий вхід мультиплексора МП1 подається ознака переносу $P_I=0$, що дозволяє передавати через мультиплексор код, що надходить із ЗП2. Якщо $R_{ij}=1$, то необхідно вибирати код із ЗП2 за адресою $N_{xi}+N_{yj}$. Якщо ж $i+1$ стає більше числа підобластей апроксимації по осі x , тобто параметри A_{ij}^1 визначаються граничними значеннями функції (при $x=a_2$), то суматор СМ1 формує сигнал переносу $P_I=1$, і на виході мультиплексора з'являється код, обраний із ЗП3, адреса комірки якого залежить лише від коду N_{yj} .

Для формування адреси комірки, у якій зберігається код параметра B_{ij}^{RT} застосовується така схема. На вхід суматора СМ2 надходить код N_{yj} , а також ознака трикутника $T(x,y)$ із СПК, і відповідно до виразу (2.71) вибирається код з комірки ЗП4 за адресою $N_{xi}N_{yj+1}$.

При цьому, якщо $T(x,y)=0$, тобто точка (x,y) знаходиться в «першому» трикутнику незалежно від виду розбивки, то вибір походить із ЗП4 за адресою $N_{xi}N_{yj}$. На керуючий вхід мультиплексора МП2 подається сигнал переносу $P_2=0$, що дозволяє передачу на вихід мультиплексора коду з ЗП2. Якщо $T(x,y)=1$, то

адреса визначається кодом $N_{xi}N_{yj+1}$. Причому, коли $j+1$ стає більше, ніж кількість підобластей апроксимації по осі Y , тобто параметри B_{ij}^{00} визначаються на межі області Ω ($y=b_2$), то суматор СМ2 формує сигнал переносу $P_2=1$, і на вихід мультиплексора МП2 дозволена передача коду з ЗП5, адреса комірки якого залежить лише від коду N_{xi} .

Відповідно до (2.78) впливає, що знак параметра B_{ij}^{RT} залежить від виду розбивки. Так при $R_{ij}=0$ знак параметра не змінюється, а при $R_{ij}=1$ знак параметра B_{ij}^{RT} необхідно змінити на протилежний. Для здійснення цієї операції служить напівсуматор, на вхід якого надходять ознака розбивки R_{ij} і записаний у ЗП4 чи в ЗП5 знак коефіцієнта: $\text{sgn}(B_{ij}^{RT})$. На виході напівсуматора отримуємо:

$$\text{sgn}'(B_{ij}^{RT}) = \text{sgn}(B_{ij}^{RT}) \oplus R_{ij} = \begin{cases} 0, & \text{якщо } \text{sgn}(B_{ij}^{RT}) = R_{ij}; \\ 1, & \text{якщо } \text{sgn}(B_{ij}^{RT}) \neq R_{ij}, \end{cases}$$

що відповідає:

$$\text{sgn}'(B_{ij}^{RT}) = (-1)^{R_{ij}} \text{sgn}(B_{ij}^{RT})$$

у виразі (2.78).

Адреса звертання до ЗП6 формує суматор СМ3. На вхід суматора надходить код N_{xi} і біт інформації з виходу напівсуматора, що відповідно до (2.72) визначає, у який з осередків i -му чи $(i+1)$ -му знаходиться необхідний код параметра C_{ij}^{RT} . Цей біт інформації виходить у результаті виконання такої операції:

$$T(x, y) \oplus R_{ij} = \begin{cases} 0, & \text{якщо } R_{ij} = T(x, y); \\ 1, & \text{якщо } R_{ij} \neq T(x, y), \end{cases}$$

що відповідає

$$1 - |T(x, y) - R_{ij}|$$

у виразі (2.72).

Інші операції аналогічні описаним для ЗП2 і ЗП3.

Отже, на виходах мультиплексорів залежно від кодів N_{xi} , N_{yj} , $N_{\Delta x}$, $N_{\Delta y}$ з'являється одна з чотирьох можливих комбінацій кодів параметрів:

$$1) \tilde{A}_{ij}^0, \tilde{B}_{ij}^{00}, \tilde{C}_{ij}^{00};$$

$$2) \tilde{A}_{ij}^0, \tilde{B}_{ij}^{01}, \tilde{C}_{ij}^{01};$$

$$3) \tilde{A}_{ij}^1, \tilde{B}_{ij}^{01}, \tilde{C}_{ij}^{01};$$

$$4) \tilde{A}_{ij}^1, \tilde{B}_{ij}^{11}, \tilde{C}_{ij}^{11}.$$

Це забезпечує правильну реалізацію виразів апроксимуючого сплайна (2.53) – (2.56)

Обрані з БП коди параметрів сплайна записуються для оперативного збереження в регістри: код параметра \tilde{A}_{ij}^R – у регістр Rг3, код параметра \tilde{B}_{ij}^{RT} – у регістр Rг4, код параметра \tilde{C}_{ij}^{RT} – у регістр Rг5.

Інші операції виконуються так само, як у пристрої, описаному в підрозділі 2.5.2.

На відміну від ФП для відтворення сплайна на трикутних підобластях апроксимації з одним видом триангуляції (з «прямою» розбивкою) розглянутий пристрій краще використовувати як спеціалізований ФП для відтворення однієї функціональної залежності.

Для відтворення функції про допомогу розглянутого ФП необхідно в його БП записати коди параметрів, для розрахунку таких параметрів відповідно до алгоритму, описаному у розділі 2.3.2, розроблена програма *TREUG2*.

Структура програми включає: основну програму, підпрограму *PROV* – для перевірки виконання умови (2.58), підпрограму *КВАНТ* – для квантування обчислених параметрів апроксимації за алгоритмом квантування, описаному в підрозділі 2.1.

Вихідні дані програми *TREUG2* містять: таблицю квантованих параметрів

апроксимації і їхніх цифрових кодів, у якій зазначено, у яке ЗП і за якою адресою необхідно занести коди параметрів; у таблиці також зазначено, яку інформацію необхідно занести в ЗП1, у якому зберігаються коди R_{ij} . Інші вихідні дані ті ж, що і для *TREUG1* (див. підрозділ 2.5.2).

2.2.3. Методи підвищення обчислюваної точності процесорів на базі сплайн перетворень

Результати розрахунків показують, що похибка квантування і похибка апроксимації істотно залежать від розрядності подання параметрів, причому можлива похибка апроксимації Q -сплайном менше, ніж сплайном з точним поданням параметрів. Очевидно, що це обумовлено вибором квантованих значень параметрів, тобто з'являється можливість на етапі розрахунку параметрів до деякої міри регулювати величину похибки. Це тим більше цінно, що для технічної реалізації цього методу не потрібно ніяких змін у схемі, а потрібно лише додатковий машинний час для здійснення розрахунку за цим методом. Отже, використовуючи малоточні елементи, є можливість домогтися більш високої точності апроксимації.

Метод виділення постійної складової параметра, що квантується, вимагає деяких змін у схемі, однак, додання несуттєві. Проте, для знакопостійних функцій метод дає зменшення похибки вдвічі.

У розділі наведено результати розрахунку для цих методів, що підтверджують їхню ефективність.

Метод виділення постійної складової параметра, що квантується. Як було показано в підрозділі 2.1, відтворення функцій пристроями з цифровим збереженням параметрів при короткій розрядній сітці слів для подання параметрів вносить істотну похибка. Загальна оцінка виникаючих під час квантування похибок описується виразами (2.4)–(2.8). Аналіз цих виразів показує, що зменшення похибки квантування можливо або за рахунок збільшення розрядності слова для подання кожного параметра, що, як відзначалося, часто не вигідно, або зменшення кожного з чисельників доданків.

Розглянемо найпростіший випадок.

Нехай задано функцію $f(x)$ на відрізку $[a; b]$. Причому на цьому відрізку вона приймає значення від f_{\min} до f_{\max} ($f(x) \in [f_{\min}; f_{\max}]$). Нехай також на цій ділянці вона апроксимована кусково-східчастою функцією

$$S(x) = \sum_{i=0}^K \xi_i(x) \cdot S_i(x),$$

де

$$S_i(x) = A_i = \frac{f_{i\max} + f_{i\min}}{2},$$

$$f_{i\min} = \min_j(f(x_{ij})), f_{i\max} = \max_j(f(x_{ij})),$$

де

$$x_{ij} = x_{i0} + j \cdot \delta x,$$

причому

$$x_{i0} = a + i \cdot h_x, (i = 0, 1, \dots, K - 1).$$

Тоді після квантування параметра оцінка похибки квантування має вигляд:

$$\Delta_{KB} = \frac{\delta A}{2} = \frac{|A|_{\max}}{2(2^l - 1)},$$

де

$$|A|_{\max} = \max_j \{|A_i|\}.$$

Очевидно, що зменшення похибки квантування можливо за рахунок зменшення величини $|A|_{\max}$.

Для того, щоб розв'язати цю задачу, необхідно виділити постійну складову параметра, що квантується.

Виділення постійної складової зробимо в такий спосіб:

$$A_c = \frac{A_{\max} + A_{\min}}{2}, \quad (2.79)$$

де

$$A_{\max} = \max_i \{A_i\},$$

$$A_{\min} = \min_i \{A_i\}.$$

При цьому крок квантування параметра A визначається в такий спосіб:

$$\delta A^* = \frac{|A|_{\max} - |A_c|}{2^l - 1} = \frac{2|A|_{\max} - |A_{\max} + A_{\min}|}{2(2^l - 1)}.$$

Можливі два випадки: $|A|_{\max} = |A_{\max}| = A_{\max}$, тобто $|A_{\max}| > |A_{\min}|$, тоді

$$|A_{\max} + A_{\min}| > 0 \text{ і } \delta A^* = \frac{2A_{\max} - A_{\max} - A_{\min}}{2(2^l - 1)} = \frac{A_{\max} - A_{\min}}{2(2^l - 1)};$$

$|A|_{\max} = |A_{\min}| = -A_{\min}$, тобто $|A_{\min}| > |A_{\max}|$, тоді $|A_{\max} + A_{\min}| < 0$ і

$$\delta A^* = \frac{-2A_{\min} + A_{\max} + A_{\min}}{2(2^l - 1)} = \frac{A_{\max} - A_{\min}}{2(2^l - 1)}.$$

Позначимо величину

$$\frac{A_{\max} - A_{\min}}{2} = A_v - \quad (2.80)$$

змінна складова параметра A .

Тоді після виділення постійної складової маємо: $\delta A^* = \frac{A_v}{2^l - 1}$.

Похибка квантування, яка дорівнює половині кроку квантування параметра, визначається як

$$\Delta_{\kappa\beta}^* = \frac{A_v}{2(2^l - 1)} = \frac{A_{\max} - A_{\min}}{4(2^l - 1)}.$$

Визначимо відносну приведену похибку:

$$\varepsilon_{\kappa\beta}^* = \frac{\Delta_{\kappa\beta}^*}{|f|_{\max}} \cdot 100\% = \frac{A_{\max} - A_{\min}}{4(2^l - 1)|f|_{\max}} \cdot 100\% = \frac{1}{2(2^l - 1)} \cdot \frac{A_v}{|f|_{\max}} \cdot 100\% = \varepsilon_{\kappa\beta} \frac{A_v}{|f|_{\max}}$$

Отже, очевидно, що якщо змінна складова параметра значно менше, ніж $|f|_{\max}$, то відносна приведена похибка істотно зменшується.

Розглянемо для прикладу наступний випадок.

Нехай $f(x)$ знакопостійна функція на відрізку $[a;b]$, і нехай $2f_{\min} = f_{\max} > 0$, тоді маємо:

$$\varepsilon_{\kappa\beta}^* = \frac{A_{\max} - A_{\max}/2}{4(2^l - 1)A_{\max}} = \frac{1}{8(2^l - 1)} = \frac{\varepsilon_{\kappa\beta}}{4},$$

оскільки $f_{\max} \approx A_{\max}$, то можлива заміна f_{\max} на A_{\max} . У цьому випадку похибка квантування зменшиться в 4 рази в порівнянні з традиційним способом квантування.

Необхідно відзначити, що традиційний спосіб квантування рівносильний розглянутому лише в тому випадку, коли:

$$|f_{\max}| = |f_{\min}|, \text{ тобто } f_{\max} = -f_{\min}.$$

Виділення постійної складової параметра, що квантується, поширюється на всі параметри.

Розглянемо функцію $f(x,y)$, що апроксимується кусково-лінійним сплайном:

$$S_{ij}(x, y) = A_{ij} + B_{ij}(x - x_i) + C_{ij}(y - y_j).$$

Для квантування параметри подано у вигляді:

$$A_{ij} = A_c + A_{vij};$$

$$B_{ij} = B_c + B_{vij};$$

$$C_{ij} = C_c + C_{vij},$$

$$\text{Тоді } S_{ij}(x, y) = [A_c + B_c(x - x_i) + C_c(y - y_j)] + [A_{vij} + B_{vij}(x - x_i) + C_{vij}(y - y_j)].$$

Розглянутий у підрозділі 2.3 пристрій реалізує вираз у других квадратних дужках. Для реалізації виразу в перших квадратних дужках вимагаються незначні зміни в схемі пристрою.

Структурна схема пристрою, що реалізує повний вираз, наведена на рис. 2.22.

Щоб реалізувати постійну складову параметра A , необхідно на вихідний суматор як ще один доданок подати відповідну частину опорної напруги. Доданки $B_c(x-x_i)$ і $C_c(y-y_j)$ реалізуються однаково. З виходів ЦАП, що формують прирости аргументів в аналоговій формі (ЦАП1 і ЦАП4), на додаткові входи суматорів з певним коефіцієнтом передачі, що відповідає множенню приросту аргументів на постійні величини B_c і C_c подаються співмножники $(x-x_i)$ і $(y-y_j)$. Отже, на вихідний суматор надходять додатково три доданки, що відповідають величинам A_c , $B_c(x-x_i)$, $C_c(y-y_j)$. Це і всі зміни в схемі, що, однак, призводять до значного зменшення похибки квантування.

Для реалізації методу виділення постійної складової параметра, що квантується, розроблено підпрограму *VCONST*.

Для роботи підпрограми *VCONST* в основній програмі має бути сформований масив *SKM*, що містить точні значення параметра, який квантується. У підпрограму передається також число елементів цього масиву.

Підпрограма *VCONST* визначає мінімальний і максимальний елементи

масиву, обчислює постійну складову P_c параметра, що квантується, змінну складову P_v параметра, обчислює нові значення параметрів за формулою:

$$P_{vij} = P_{ij} - P_c,$$

де P_{vij} – нове значення параметра P_{ij} після виділення постійної складової; P_{ij} – точне значення параметра апроксимації. Змінна складова P_v використовується в підпрограмі КВАНТ для визначення кроку квантування параметра.

Підпрограма *VCONST* випробувана в складі програми *TREUG1*. Для цього в програмі *TREUG1* були введені операції щодо підготовки масиву *SKM* для кожного параметра.

За допомогою програми *TREUG1* і підпрограми *VCONST* проведено дослідження результатів апроксимації лінійним Q -сплайном двох змінних на трикутних підобластях апроксимації з виділенням постійної складової параметра, що квантується, ряду функцій. Для останньої графі використано дані, отримані під час застосування програми *TREUG1* при традиційному способі квантування, що наведені в табл. 2.1.

Для методу виділення постійної складової параметра, що квантується, практично у всіх випадках спостерігається істотне зменшення похибки квантування. Це зменшення відбувається вдвічі. Дійсно, основний вплив на похибку квантування робить крок квантування вільного коефіцієнта A , тому що у вузлових точках значення Q -сплайна визначається тільки значенням квантованого параметра \tilde{A} . Отже, зменшення величини кванта вільного коефіцієнта призведе до зменшення похибки у вузлових точках і вплине на повну похибку квантування.

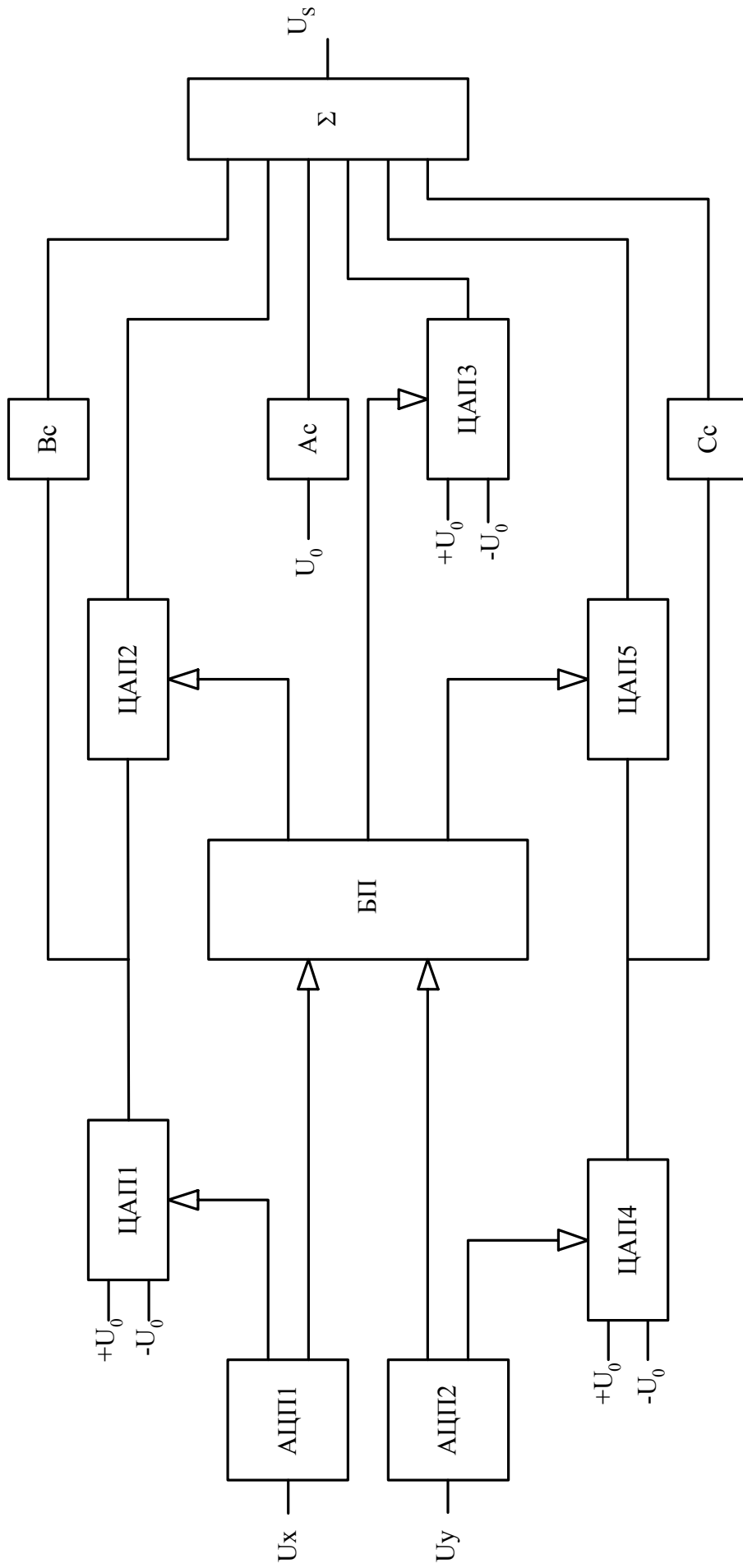


Рисунок 2.2.22 – Структурна схема ФП-2, що реалізує Q-сплайн з виділенням постійної складової параметра, що квантується

Практично для всіх прикладів функція приймає значення від 0 до деякої величини f_{\max} , за винятком функцій $f(x, y) = \ln(x; y)$ і $f(x, y) = \sin(x - y)$. При цьому вільний коефіцієнт також приймає значення від 0 у вузловій точці (0;0) і до деякої величини $|A|_{\max}$. Після виділення постійної складової крок квантування цього параметра зменшиться вдвічі. Крім того, усі наведені приклади стосуються функцій, що монотонно змінюються на розглянутих областях визначення, і, отже, параметри B і C приймають значення відповідно від $B_{\min} \geq 0$ і до $B_{\max} > 0$ і від $C_{\min} \geq 0$ до $C_{\max} > 0$ (для функції $f(x, y) = e^x \cos y$ від $B_{\min} < 0$ до $B_{\max} \leq 0$ і від $C_{\min} < 0$ до $C_{\max} \leq 0$). Виділення постійної складової для цих параметрів також призведе до зменшення кроку квантування параметрів як мінімум у два рази. Отже, і загальна похибка квантування має зменшитися вдвічі.

Інше спостерігається для функції $f(x, y) = \sin(x_1 - y)$ на інтервалі $[0; \frac{\pi}{2}] \cdot [0; \frac{\pi}{2}]$. На цьому інтервалі функція приймає значення від -1 до 1, тобто $f_{\max} = -f_{\min}$ і, отже, параметр A приймає значення $[-1; 1]$ (верхня межа може бути трохи менше, особливо для малого числа підобластей апроксимації). Виділення постійної складової для цього параметра призведе до несуттєвого зменшення кроку квантування, або крок квантування не зміниться. Для параметрів B і C виділення постійної складової призведе до зменшення кроку квантування вдвічі, тому що ці параметри змінюються від 0 до деякої величини $B_{\max} > 0$ і $C_{\max} > 0$. Однак, зменшення кроку квантування параметрів B і C не призводить до зменшення похибки квантування у вузлових точках. Тому в даному випадку, тобто для знакозмінної функції, коли $f_{\max} = -f_{\min}$ цей спосіб не має переваги, що підтверджують розрахунки для функції $f(x, y) = \sin(x - y)$ на інтервалі $\Omega = [0; \frac{\pi}{2}] \cdot [0; \frac{\pi}{2}]$.

Метод вибору оптимальних квантованих значень параметрів.

Розрахунок параметрів Q -сплайнів за допомогою традиційного алгоритму, тобто вибору найближчих дозволених рівнів для подання параметра, показує, що фактична похибка апроксимації Q -сплайном буває менше похибки апроксимації сплайном з точним поданням параметрів.

Необхідно також відзначити, що похибка квантування, будучи величиною випадковою, менше оцінок, обумовлених формулами (2.4) – (2.8). Оцінка виходить завищеною через малу імовірність того, що похибки подання кожного параметра в даній підобласті дорівнюватимуть половині кроку квантування з однаковим знаком. Однак ці формули забезпечують те, що похибка квантування ніколи не буде більше цих оцінок.

При традиційному методі квантування похибка квантування оцінюється величиною, наприклад, для кусково-лінійного сплайна однієї змінної:

$$\Delta_{кв} = \frac{\delta q^{[0]}}{2} + \frac{\delta q^{[1]}}{2}(x - x_i).$$

Аналіз цієї формули показує, що на лівому кінці інтервалу, тобто в точці x_i , похибка дорівнює $\frac{\delta q^{[0]}}{2}$. Цю похибку можна зменшити, тільки зменшуючи крок квантування параметра $q^{[0]}$, а подальше зменшення похибки квантування за рахунок збільшення розрядної сітки веде до збільшення апаратних витрат і подорожчання пристрою.

В міру переміщення від точки x_i до точки x_{i+1} при традиційному способі похибка квантування може зростати, якщо похибка подання обох параметрів найближчими дозволеними рівнями має однаковий знак, тому що зростає множник $(x-x_i)$ і досягати свого максимуму в точці x_{i+1} , коли $x_{i+1} - x_i = h_x$ і стає рівною:

$$\Delta_{кв} = \frac{\delta q^{[0]}}{2} + \frac{\delta q^{[1]}}{2} \cdot h_x.$$

Подамо кожен параметр у вигляді:

$$q_i = \tilde{q}_i + \Delta\tilde{q}_i,$$

де \tilde{q}_i – квантоване значення параметра; $\Delta\tilde{q}_i$ – похибка подання параметра q_i квантованим значенням \tilde{q}_i .

Тоді для i -ї підобласті апроксимації маємо:

$$\Delta_{kvi}(x) = \Delta\tilde{q}_i^{[0]} + \Delta\tilde{q}_i^{[1]} \cdot (x - x_i).$$

У точці x_i похибка приймає значення $\Delta\tilde{q}_i^{[0]}$ в точці x_{i+1} – значення:

$$\Delta_{kvi}(x_{i+1}) = \Delta\tilde{q}_i^{[0]} + \Delta\tilde{q}_i^{[1]} \cdot h_x.$$

Необхідно відзначити, що величина $\Delta\tilde{q}$ може приймати позитивні і негативні значення, тому що

$$\Delta\tilde{q}_i \in \{q_i - \tilde{q}_i^l; q_i - \tilde{q}_i^{l-1}\},$$

де \tilde{q}_i^{l-1} – найближче менше значення з розріджених рівнів; \tilde{q}_i^l – найближче більше значення з дозволених рівнів.

Але в цьому випадку природно вибрати такі значення \tilde{q}_i , щоб величини $\Delta\tilde{q}_i^{[0]}$ і $\Delta\tilde{q}_i^{[1]} \cdot h_x$ компенсували одна одну, тобто були з протилежними знаками.

Отже, виникає задача вибору таких квантованих значень параметрів, щоб похибка квантування при апроксимації Q -сплайном була мінімальною.

Розглянемо розв'язання цієї задачі для кусково-лінійної апроксимації Q -сплайном функції однієї змінної.

Залежно від вибору одного з найближчих рівнів \tilde{q}_i^{l-1} чи \tilde{q}_i^l , між якими знаходиться точне значення параметра $\tilde{q}_i^{l-1} \leq \tilde{q}_i \leq \tilde{q}_i^l$, можливі різні величини похибки, внесеної квантуванням (для зменшення кількості індексів позначимо

$$\begin{aligned} q_i^{[0]} &= A_i, q_i^{[1]} = B_i; \\ \Delta_{\kappa\tilde{v}i}^1(x) &= \Delta\tilde{A}_i^l + \Delta\tilde{B}_i^m \cdot (x - x_i); \\ \Delta_{\kappa\tilde{v}i}^2(x) &= \Delta\tilde{A}_i^l + \Delta\tilde{B}_i^{m-1} \cdot (x - x_i); \\ \Delta_{\kappa\tilde{v}i}^3(x) &= \Delta\tilde{A}_i^{l-1} + \Delta\tilde{B}_i^m \cdot (x - x_i); \\ \Delta_{\kappa\tilde{v}i}^4(x) &= \Delta\tilde{A}_i^{l-1} + \Delta\tilde{B}_i^{m-1} \cdot (x - x_i). \end{aligned}$$

При цьому необхідно вибрати такі значення параметрів, що квантуються, щоб забезпечити $\Delta_{\kappa\tilde{v}i} = \min_j \left\{ \left| \Delta_{\kappa\tilde{v}i}^j \right| \right\}$.

Як було відзначено, максимуму за абсолютною величиною $\Delta_{\kappa\tilde{v}i}$ може досягати на кінцях інтервалу $[x_i; x_{i+1}]$, тобто в точках x_i, x_{i+1} . Розглянемо ці випадки:

$$\begin{aligned} \Delta_{\kappa\tilde{v}i}^1(x_i) &= \Delta\tilde{A}_i^l; \\ \Delta_{\kappa\tilde{v}i}^1(x_{i+1}) &= \Delta\tilde{A}_i^l + \Delta\tilde{B}_i^m \cdot h_x; \\ \Delta_{\kappa\tilde{v}i}^2(x_i) &= \Delta\tilde{A}_i^l; \\ \Delta_{\kappa\tilde{v}i}^2(x_{i+1}) &= \Delta\tilde{A}_i^l + \Delta\tilde{B}_i^{m-1} \cdot h_x; \\ \Delta_{\kappa\tilde{v}i}^3(x_i) &= \Delta\tilde{A}_i^{l-1}; \\ \Delta_{\kappa\tilde{v}i}^3(x_{i+1}) &= \Delta\tilde{A}_i^{l-1} + \Delta\tilde{B}_i^m \cdot h_x; \\ \Delta_{\kappa\tilde{v}i}^4(x_i) &= \Delta\tilde{A}_i^{l-1}; \\ \Delta_{\kappa\tilde{v}i}^4(x_{i+1}) &= \Delta\tilde{A}_i^{l-1} + \Delta\tilde{B}_i^{m-1} \cdot h_x. \end{aligned}$$

Отже, необхідно вибрати зі значень \tilde{A}_i^{l-1} , \tilde{A}_i^l , \tilde{B}_i^{m-1} , \tilde{B}_i^m таку пару значень \tilde{A}_i^* , \tilde{B}_i^* щоб забезпечити виконання такої умови:

$$\begin{aligned} \Delta_{\kappa\epsilon i} = \min \{ & \max(|\Delta\tilde{A}_i^l|; |\Delta\tilde{A}_i^l + \Delta\tilde{B}_i^m \cdot h_x|), \\ & \max(|\Delta\tilde{A}_i^l|; |\Delta\tilde{A}_i^l + \Delta\tilde{B}_i^{m-1} \cdot h_x|); \max(|\Delta\tilde{A}_i^{l-1}|; |\Delta\tilde{A}_i^{l-1} + \Delta\tilde{B}_i^m \cdot h_x|); \\ & \max(|\Delta\tilde{A}_i^{l-1}|; |\Delta\tilde{A}_i^{l-1} + \Delta\tilde{B}_i^{m-1} \cdot h_x|). \end{aligned}$$

При обраних параметрах \tilde{A}_i^* , \tilde{B}_i^* забезпечуватиметься мінімум похибки квантування.

Розглянуте розв'язання задачі веде до зменшення похибки квантування. Однак, розглянемо загальну похибку апроксимації:

$$\tilde{\Delta} = |\Delta_{\mathcal{M}} + \Delta_{\kappa\epsilon}|,$$

де $\Delta_{\mathcal{M}} = S(x) - f(x)$ – методична похибка; $\Delta_{\kappa\epsilon} = \tilde{S}(x) - f(x)$ – похибка квантування.

Як, $\Delta_{\mathcal{M}}$ так і $\Delta_{\kappa\epsilon}$ можуть приймати позитивні і негативні значення. Через те, що, вибираючи параметри \tilde{A}_i^* і \tilde{B}_i^* , можна керувати в деяких межах величиною і знаком $\Delta_{\kappa\epsilon}$, то можна змінити умову задачі і знаходити $\tilde{\Delta}_i$ в такий спосіб:

$$\begin{aligned} \tilde{\Delta}_i = \min \{ & \max_j \{ |\tilde{A}_i^l + \tilde{B}_i^m (x_{ij} - x_{i0}) - f(x_{ij})| \}; \\ & \max_j \{ |\tilde{A}_i^l + \tilde{B}_i^{m-1} (x_{ij} - x_{i0}) - f(x_{ij})| \}; \\ & \max_j \{ |\tilde{A}_i^{l-1} + \tilde{B}_i^m (x_{ij} - x_{i0}) - f(x_{ij})| \}; \end{aligned}$$

$$\max_j \left\{ \left| \tilde{A}_i^{l-1} + \tilde{B}_i^{m-1} (x_{ij} - x_{i0}) - f(x_{ij}) \right| \right\},$$

де $x_{ij} = x_{i0} + j \cdot \delta x$, $j = 0, 1, 2, \dots$, причому $x_{ij} \in [x_i; x_{i+1}]$.

Вибір \tilde{A}_i^* і \tilde{B}_i^* відповідно до цієї умови може збільшити похибку квантування $|\Delta_{\kappa\epsilon}|$, однак при цьому загальна похибка апроксимації Q -сплайном $\tilde{\Delta}$ може бути менше, ніж похибка апроксимації тим самим методом, але з точним поданням параметрів Δ_m .

Для підтвердження вищенаведених міркувань розглянемо рисунки, геометрично пояснюючі викладені висновки. На рис. 2.23 наведено графіки функцій $\Delta_{\kappa\epsilon}^j(x) = \tilde{S}_i(x) - S_i(x)$.

Функція, що апроксимується, $f(x) = \sqrt{x}$, $x \in [0; 3]$, причому вся область апроксимації розбита на 8 підобластей. На рисунку наведено ділянку $[0,375; 0,736]$. Розрядність подання параметрів $l^{[\alpha]} = 8(\alpha = \overline{0,1})$. Вибираємо $\tilde{A}_i^* = \tilde{A}_i^{l-1}$.

Похибка квантування при різному виборі квантованих значень параметрів $\tilde{B}_i^* = \tilde{B}_i^m$, тому що в цьому випадку забезпечується мінімальне значення похибки квантування $|\Delta_{\kappa\epsilon i}|$. А саме: $|\Delta_{\kappa\epsilon i}^3| = |\Delta_{\tilde{A}_i}^{l-1}| = 2,42 \cdot 10^{-3}$.

На рис. 2.24 зображено графіки функцій $\Delta_{mi}(x) = |S_i(x) - f(x)|$, $\tilde{\Delta}_i^j(x) = |\tilde{S}_i^j(x) - f(x)|$.

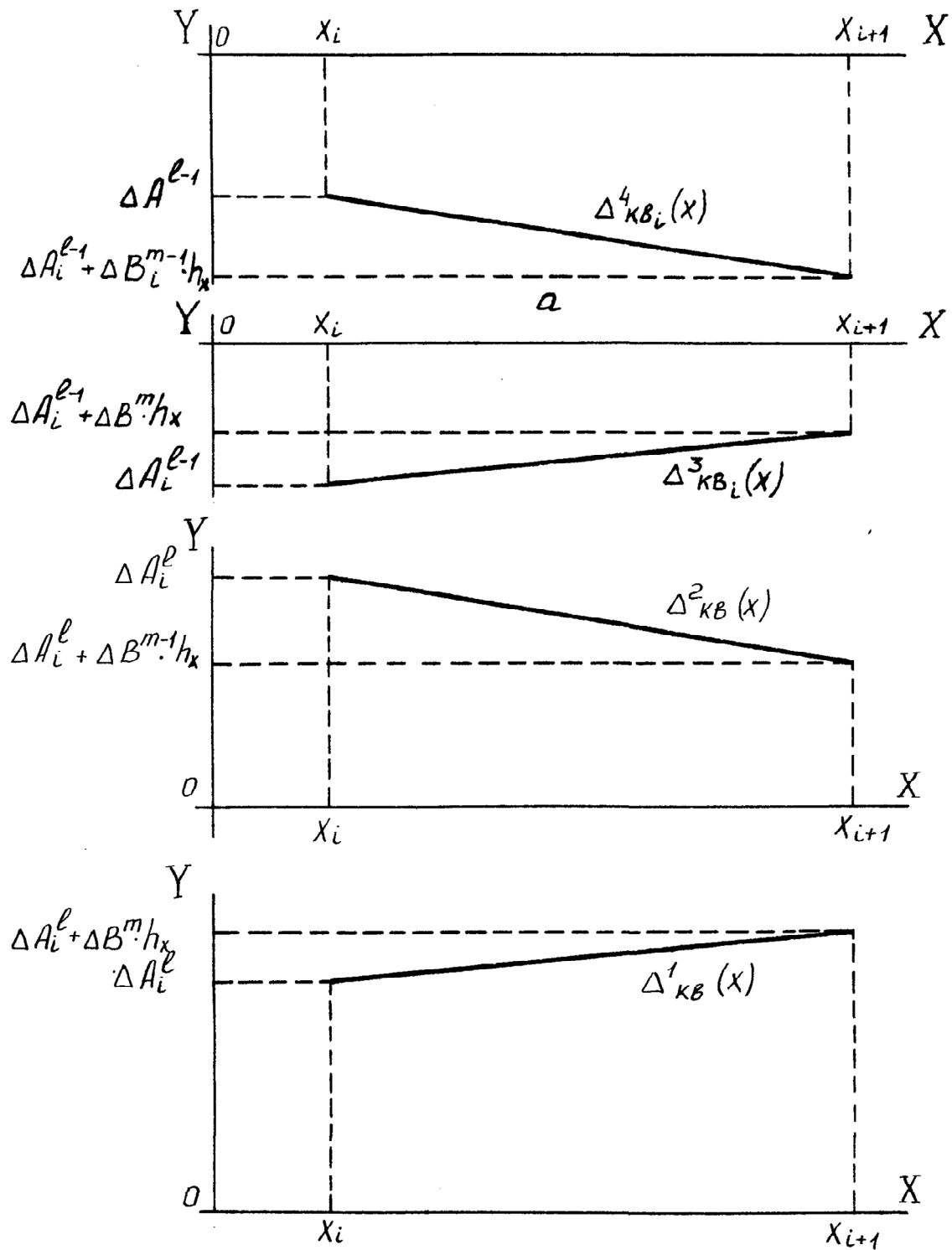


Рисунок 2.23 – Похибка квантування при різних способах вибору значень квантованих параметрів

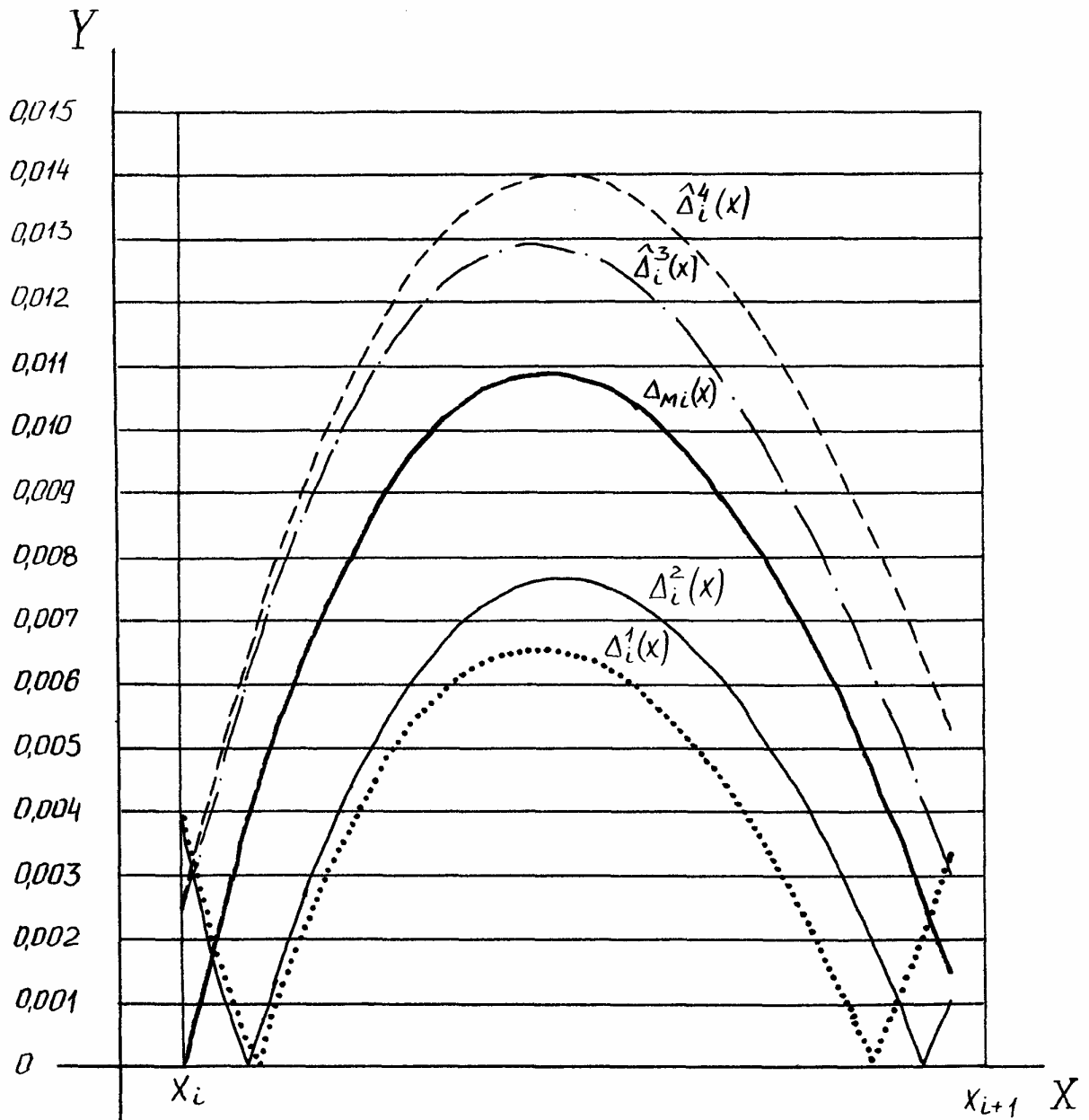


Рисунок 2.24 – Похибка апроксимації сплайнами при різному виборі параметрів

Аналіз графіків на рис. 2.24 показує, що необхідно вибрати параметри $\tilde{A}_i^* = \tilde{A}_i^l$, $\tilde{B}_i^* = \tilde{B}_i^m$, що забезпечують найменшу похибку апроксимації Q -сплайном. З графіків видно, що в даному випадку похибка апроксимації Q -сплайном буде менше, ніж при апроксимації сплайном з точним поданням параметрів. Отже, більш простий спосіб вибору квантованих значень параметрів за мінімумом похибки квантування не забезпечує найкращого наближення.

Недоліком запропонованого способу є істотне збільшення обсягу обчис-

лень. У випадку кусково-лінійної апроксимації функції однієї змінної при виборі квантованих параметрів обсяг обчислень зростає в 4 рази. Збільшення обсягу обчислень можна оцінити величиною 2^α , де α – кількість параметрів апроксимації.

Вище розглянутий випадок апроксимації функції однієї змінної лінійним Q -сплайном.

Запишемо умови для всіх розглянутих у роботі випадків:

1. Для параболічного Q -сплайна однієї змінної
- мінімізація похибки квантування

$$\Delta_{kvi} = \min_{\forall \lambda, \mu, \rho} \left\{ \max \left\{ \left| \Delta \tilde{A}_i^\lambda \right|; \left| \Delta \tilde{A}_i^\lambda + \Delta \tilde{B}_i^\mu \cdot h_x + \Delta \tilde{C}_i^\rho \cdot h_x^2 \right| \right\} \right\},$$

де $\lambda \in \{l-1; l\}$, $\mu \in \{m-1; m\}$, $\rho \in \{r-1; r\}$ – номери найближчих меншого і більшого дозволених рівнів для кожного коефіцієнта:

- мінімізація похибки апроксимації

$$\tilde{\Delta}_i = \min_{\forall \lambda, \mu, \rho} \left\{ \max_j \left\{ \left| \Delta \tilde{A}_i^\lambda + \Delta \tilde{B}_i^\mu \cdot (x_{ij} - x_{i0}) + \Delta \tilde{C}_i^\rho \cdot (x_{ij} - x_{i0})^2 - f(x_{ij}) \right| \right\} \right\}.$$

2. Для кубічного Q -сплайна однієї змінної – мінімізація похибки квантування:

$$\Delta_{eai} = \min_{\forall \lambda, \mu, \rho, \tau} \left\{ \max \left\{ \left| \Delta \tilde{A}_i^\lambda \right|; \left| \Delta \tilde{A}_i^\lambda + \Delta \tilde{B}_i^\mu \cdot h_x + \Delta \tilde{C}_i^\rho \cdot h_x^2 + \Delta \tilde{D}_i^\tau \cdot h_x^3 \right| \right\} \right\};$$

- мінімізація похибки апроксимації:

$$\tilde{\Delta}_i = \min_{\forall \lambda, \mu, \rho, \tau} \left\{ \max_j \left\{ \left| \Delta \tilde{A}_i^\lambda + \Delta \tilde{B}_i^\mu \cdot (x_{ij} - x_{i0}) + \Delta \tilde{C}_i^\rho \cdot (x_{ij} - x_{i0})^2 + \tilde{D}_i^\tau (x_{ij} - x_{i0})^3 - f(x_{ij}) \right| \right\} \right\}.$$

3. Для білінійного Q -сплайна двох змінних на прямокутних підобластях апроксимації

– мінімізація похибки квантування:

$$\Delta_{\kappa\epsilon ij} = \min_{\forall \lambda, \mu, \rho, \tau} \left\{ \max \left\{ \left[\left| \Delta \tilde{A}_{ij}^{\lambda} \right|; \left| \Delta \tilde{A}_{ij}^{\lambda} + \Delta \tilde{B}_{ij}^{\mu} \cdot h_x \right|; \left| \Delta \tilde{A}_{ij}^{\lambda} + \Delta \tilde{C}_{ij}^{\rho} \cdot h_y \right|; \right] \right\} \right\};$$

– мінімізація похибки апроксимації:

$$\tilde{\Delta}_{ij} = \min_{\forall \lambda, \mu, \rho, \tau} \left\{ \max_{u, s} \left\{ \left[\left| \Delta \tilde{A}_{ij}^{\lambda} + \Delta \tilde{B}_{ij}^{\mu} \cdot (x_{is} - x_{i0}) + \Delta \tilde{C}_{ij}^{\rho} \cdot (y_{ju} - y_{j0}) \right| \right] \right\} \right\}.$$

2. Для лінійного Q -сплайна двох змінних на трикутних підобластях апроксимації (для кожної трикутної підобласті) – мінімізація похибки квантування:

$$\Delta_{\epsilon\hat{a}ij} = \min_{\forall \lambda, \mu, \rho} \left\{ \max \left\{ \left[\left| \Delta \tilde{A}_{ij}^{\lambda} + (1 - T(x, y)) \cdot \Delta \tilde{B}_{ij}^{\mu T} \cdot h_x + T(x, y) \cdot \Delta \tilde{C}_{ij}^{\rho T} \cdot h_y \right|; \right] \right\} \right\}.$$

Ознака $T(x, y)$ вказує для якого з трикутників йде визначення похибки.

– мінімізація похибки апроксимації:

$$\tilde{\Delta}_{ij} = \min_{\forall \lambda, \mu, \rho} \left\{ \max_{s, u} \left\{ \left[\left| \Delta \tilde{A}_{ij}^{\lambda} + \Delta \tilde{B}_{ij}^{\mu T} \cdot (x_{is} - x_{i0}) + \Delta \tilde{C}_{ij}^{\rho T} \cdot (y_{ju} - y_{j0}) - f(x_{is}; y_{ju}) \right| \right] \right\} \right\}.$$

Метод вибору оптимальних значень квантованих параметрів реалізований

за допомогою підпрограми *OPKB*, що включена до складу програми *TREUG1*.

Виклик підпрограми проводиться після визначення квантованих значень параметрів традиційним способом.

Підпрограма *OPKB* для кожної трикутної підобласті апроксимації здійснює обчислення найближчих квантованих значень кожного параметра апроксимації, похибки подання кожного параметра апроксимації найближчими великим і меншим значеннями, обчислює похибку квантування при різних комбінаціях квантованих значень параметрів, робить вибір такої комбінації, що забезпечує мінімальне значення похибки квантування.

Щоб здійснити мінімізацію похибки апроксимації, розроблено підпрограму-функцію *EPMAX*, що служить для обчислення похибки апроксимації, а не похибки квантування. Підпрограма-функція *EPMAX* здійснює розрахунок похибки апроксимації для кожної комбінації квантованих значень параметрів у всіх перевірних точках (сітка Δ_{xy}). Потім передає це значення в підпрограму *OPKB*, що працює з ним так само, як з похибкою квантування, тобто визначає комбінацію квантованих значень параметрів, що забезпечують мінімум похибки.

Під час розробки підпрограми *OPKB* виникають труднощі визначення комбінації квантованих значень параметрів, пов'язані з тим, що параметр A_{ij} є загальним для двох трикутних підобластей i , отже, може виявитися, що вимагаються різні квантовані значення цього параметра для забезпечення мінімальної похибки в кожній трикутній підобласті. У зв'язку з цим у підпрограмі передбачено такий алгоритм вибору оптимальних значень параметрів:

1. Для A_{ij}^λ ($\lambda \in \{l-1; l\}$), найближчих дозволених значень, обчислюються похибки апроксимації $\tilde{A}_{ij}^{[T]}$ (\tilde{A}_{ij}^λ) для кожної трикутної підобласті ($T(x,y) = \overline{0,1}$) у відповідності з таким виразом:

$$\tilde{\Delta}_{ij}^{[T]}(\tilde{A}_{ij}^{\lambda}) = \min_{\forall \mu, \rho} \left\{ \max_{s, y} \left\{ \left| \Delta \tilde{A}_{ij}^{\lambda} + \Delta \tilde{B}_{ij}^{\mu [T]} \cdot (x_{is} - x_{i0}) + \Delta \tilde{C}_{ij}^{\rho [T]} \cdot (y_{ju} - y_{j0}) \right| \right\} \right\},$$

де $\mu \in \{m-1; m\}$;

$$\rho \in \{r-1; r\};$$

$$x_{is} = x_{i0} + s \cdot \delta x;$$

$$y_{ju} = y_{j0} + u \cdot \delta y.$$

При цьому відбувається вибір значень параметрів $\tilde{B}_{ij}^{[T]}(\tilde{A}_{ij}^{\lambda})$, $\tilde{C}_{ij}^{[T]}(\tilde{A}_{ij}^{\lambda})$ які забезпечують цю похибку.

2. Вибирається значення \tilde{A}_{ij}^* , що забезпечує

$$\tilde{\Delta}_{ij} = \min \left\{ \max_{T=0,1} \left\{ \tilde{\Delta}_{ij}^{[T]}(\tilde{A}_{ij}^l) \right\}; \max_{T=0,1} \left\{ \tilde{\Delta}_{ij}^{[T]}(\tilde{A}_{ij}^{l-1}) \right\} \right\}.$$

При цьому буде визначена комбінація параметрів \tilde{A}_{ij}^* , $\tilde{B}_{ij}^{[0]*}$, $\tilde{C}_{ij}^{[0]*}$, $\tilde{B}_{ij}^{[1]}$, $\tilde{C}_{ij}^{[1]}$, що забезпечує мінімум похибки на прямокутній підобласті.

Отримані дані чітко свідчать про істотний вигреш у точності, особливо для коротких розрядних сіток 8–12 розрядів (вірніше, для тих випадків, коли похибка квантування стає порівнянною з похибкою апроксимації сплайном з точним поданням параметрів), методу мінімізації похибки апроксимації. Однак, як відзначалося, цей метод призводить до збільшення похибки квантування і, отже, до більшого розходження Q -сплайна і сплайна з точним поданням параметрів. Такий метод придатний для забезпечення кращого рівномірного наближення функції Q -сплайном, але не дуже вдалий за необхідності забезпечення мінімальної похибки у вузлових точках. В останньому випадку бажане застосування методу мінімізації похибки квантування.

2.3. Розробка архітектури обчислювачів для колірних перетворень зображень

2.3.1 Алгоритми колориметричної обробки ТВ зображень

Людське око є кінцевою ланкою в системах передачі візуальної інформації. Тому побудова телевізійних систем не може здійснюватися без урахування властивостей зору. Око сприймає світлове випромінювання в дуже широкому інтервалі яскравостей. Відношення максимальної яскравості B_{max} , що відповідає болючому відчуттю, до мінімального B_{min} , ще помітної оком, досягає 10^{11} - 10^{12} . Цей діапазон яскравостей охоплюється не одночасно, тому що око щоразу адаптується до тієї чи іншої середньої яскравості. Зорове відчуття яскравості неможливо безпосередньо виміряти, тому його оцінюють непрямим методом. Він полягає в тому, що при розгляді дотичних напівкіл яскравістю B і $B+\Delta B$, розташованих у центрі поля з яскравістю B_ϕ , що визначає рівень адаптації ока, виникає, відповідно, зорове відчуття A і $A+\Delta A$. Відповідно до закону Вебера-Фехнера

$$\Delta A = k \frac{\Delta \hat{A}}{\hat{A}}, \quad (2.81)$$

де k – коефіцієнт пропорційності. Замінивши збільшення диференціалами і після інтегрування рівняння, отримаємо:

$$\hat{A} = C \ln \hat{A} + \tilde{N}, \quad (2.82)$$

де C – постійна інтегрування. З виразу (2.82) випливає, що зорове відчуття пропорційне логарифму яскравості.

Однак така залежність спостерігається у відносно вузькому діапазоні яскравостей, і не є точною, тому що не враховує яскравість фону B_ϕ , до якої адаптується око. Експериментально зняті залежності показали, що при даній яскравості фону B_ϕ , контрастна чутливість ока максимальна для деталей, яскравість яких близька до B_ϕ і зростає зі збільшенням B_ϕ . Під контрастною чутливістю розуміється відношення яскравості B до граничної різниці яскравостей $\Delta B_{пор}$, що створює відчуття граничного розрізнення ока $\Delta A_{пор}$.

З вищенаведених міркувань випливає, що для створення найбільш комфортних для спостерігача умов сприйняття зображення в цілому й окремих його ділянках необхідно забезпечити управління яскравістю і контрастом зображення, а також кольором фону, на якому воно наведено.

Дуже важливо враховувати, що в зоровому апараті поряд з яскравісною адаптацією існує і колірна адаптація, що виражається в тому, що колір, на який адаптується око, як би вицвітає. Це призводить до зміни відчуття кольору в результаті попереднього впливу на око інших кольорів (послідовний колірний контраст) чи до удаваної зміни кольору деталі при зміні кольору фону (одночасний колірний контраст). Характерно і те, що в будь-яких умовах висвітлення білі деталі завжди легко пізнаються, тому що вони найяскравіші, а вже стосовно них око оцінює всі інші кольори. Таке явище називається контрастністю кольору чи виправленням на висвітлення.

Згідно з 1-м законом колориметрії (науки про вимір і кількісний вираз кольору), будь-які чотири кольори знаходяться в лінійній залежності, однак існує необмежена кількість комбінацій із трьох кольорів, що виявляються лінійно незалежними (тобто, змішуючи два з таких кольорів, ні за яких умов не можна одержати третій). Отже, будь-який колір може виражатися однією з комбінацій трьох лінійно незалежних кольорів, що у даній колориметричній системі називаються основними. При цьому колір суміші не залежить від природи компонентів, що змішуються (3-й закон колориметрії).

У 1931 р. Міжнародна комісія з освітлення (МКО) як основні кольори стандартизувала червоний R (700 нм), зелений G (546,1 нм) і синій B (435,8 нм). У такій колориметричній системі зображення може подаватися у вигляді трьох таблиць, кожна з яких відповідає одному з трьох колірних каналів (червоному, зеленому чи синьому). Кількість рядків в одній такій таблиці дорівнює кількості рядків, у які здійснюється розгорнення зображення в даній системі обробки, а кількість стовпців, відповідно, кількості точок в одному рядку. Кожна комірка таблиці містить інтенсивність даної колірної складової у

відповідній точці зображення. У загальному вигляді така таблиця може подаватися деякою функцією $E = E(x,y)$, що виражає залежність інтенсивності колірної складової E від координат (x,y) точки на екрані. Область визначення такої функції обмежена розмірами кадру, а область припустимих значень належить інтервалу від нуля до одиниці, (нуль означає, що дана колірна складова відсутня у даній точці, а одиниця – що колірна складова має максимальну інтенсивність для даного типу обробної апаратури). При такому поданні яскравість зображення в даній точці дорівнює:

$$Y(x, y) = 0,3E_R + 0,59E_G + 0,11E_B, \quad (2.83)$$

де $E_R = E_R(x,y)$ – інтенсивність червоної колірної складової;

$E_B = E_B(x,y)$ – інтенсивність червоної колірної складової;

$E_G = E_G(x,y)$ – інтенсивність червоної колірної складової.

Тут чисельні значення коефіцієнтів обрані з урахуванням характеристик прийнятого умовного білого кольору D_{6500} (джерела з колірною температурою 6500 К, що відповідає випромінюванню денного хмарного неба) і колірних координат у колориметричній системі XYZ люмінофорів основних кольорів під час відтворення кольорових зображень.

Як відомо, передавальні і приймальні телевізійні трубки, відео-магнітофони різного типу і інші є джерелами нелінійних перекручувань зображення. У загальному випадку, для збереження подібності переданого і відтвореного зображень необхідно забезпечити пряму пропорційність між яскравістю $Y_{об}$ елементів переданого зображення об'єкта і яскравістю $Y_з$ відтвореного зображення об'єкта:

$$Y_з = mY_{об}, \quad (2.84)$$

де m – деякий коефіцієнт пропорційності. За необхідності підкреслення чи згладжування контрасту зони вимога (2.84) може і не дотримуватися.

Як уже було зазначено, напруга $U_{вих}$ на виході передавальної трубки пов'язана з яскравістю об'єкта нелінійною залежністю виду:

$$U_{вих} = m_1 Y_{об}^{\gamma_1}, \quad (2.85)$$

де m_1 – деякий коефіцієнт пропорційності, γ_1 – коефіцієнт нелінійності передавальної трубки. Напряга $U_{вх}$ надходить каналом зв'язку на вхід системи обробки, що деяким чином перетворить його в напрягу, що надходить на відтворюючу ЕПТ. Нехай ця залежність має вигляд:

$$U_{\text{вх}} = m_1 U_{\text{вх}}^{\gamma_1} = m_1 m_2 U_{\text{вх}}^{\gamma_2} = m_1 m_2 m_3 U_{\text{вх}}^{\gamma_3}, \quad (2.86)$$

де m_2 – коефіцієнт пропорційності, γ_2 – коефіцієнт нелінійності тракту обробки. Яскравість елементів відтвореного зображення пов'язана нелінійною залежністю з напрягою $U_{\text{сприйн}}$. З урахуванням (2.85) і (2.86) запишемо залежність яскравості відтвореного зображення від яскравості об'єкта:

$$Y_{\text{с}} = m_3 U_{\text{сприйн}}^{\gamma_3} = m_3 m_2^{\gamma_2} m_1^{\gamma_1 \gamma_2} Y_{\text{об'єкт}}^{\gamma_1 \gamma_2 \gamma_3} = m Y_{\text{об'єкт}}^{\gamma}, \quad (2.87)$$

де m – коефіцієнт передачі системи, γ – коефіцієнт нелінійності системи. Для виконання умови (2.84) необхідно, щоб

$$\gamma = \gamma_1 \gamma_2 \gamma_3 = 1. \quad (2.88)$$

Необхідно розуміти, що коефіцієнти нелінійності передавальної γ_1 і приймальної γ_3 апаратури вважаються відомими і не підлягають коректуванню. Тоді забезпечення виконання співвідношення (2.88) здійснюється за рахунок настроювання коефіцієнта нелінійності γ_2 тракту обробної апаратури і реалізації в ній передавальної характеристики виду (2.86). Цю операцію виконує блок, що називається гамма-коректором. За необхідності відхилення від умови (2.84) (наприклад, для підкреслення контрасту переважно світлих чи темних місць) значення γ_2 може не задовольняти співвідношення (2.88).

Як відомо, у цифровій апаратурі напруги подаються у вигляді відліків, формованих аналого-цифровими перетворювачами (АЦП) з лінійною передавальною характеристикою. Так, напруга відеосигналу $U_{\text{вих}}$ з виходу передавальної апаратури каналом зв'язку надходить на вхід системи обробки. Пристроєм, що декодує, з цієї напруги виділяються сигнали кольорних складових. Ці сигнали надходять на АЦП, де квантуються і записуються у відповідні комірки оперативної пам'яті системи. Раніше уведена функція $E(x,y)$

описує зображення в пам'яті одного з колірних каналів. Відеопроцесори обробної системи визначають послідовність читання зображення з оперативної пам'яті, перетворюючи функцію $E(x,y)$. Зображення, що зчитується далі з пам'яті (подано у вигляді відліків інтенсивностей колірних складових $ERex(x,y)$, $EGex(x,y)$, $EBex(x,y)$), надходить на гамма-коректор, що реалізує для кожного каналу залежність вигляду:

$$\begin{aligned} E_R(x,y) &= m_R (E_{RBx}(\tilde{o}, \hat{o}))^{\gamma_R}; \\ E_G(x,y) &= m_G (E_{GBx}(\tilde{o}, \hat{o}))^{\gamma_G}; \\ E_B(x,y) &= m_B (E_{BBx}(\tilde{o}, \hat{o}))^{\gamma_B}, \end{aligned} \quad (2.89)$$

де m_R, m_G, m_B , – коефіцієнти передачі гамма-коректора у відповідному колірному каналі, $\gamma_R, \gamma_G, \gamma_B$, – коефіцієнти нелінійності передачі у відповідному колірному каналі цього гамма-коректора. Отже, розглянуті раніше принципи гамма-корекції поширюються на кожен колірний канал.

При апаратній реалізації цифрового гамма-коректора ставиться задача синтезу функції вигляду (2.89). Крім того, необхідно забезпечити можливість зміни коефіцієнта нелінійності в залежності від змісту і якості оброблюваного зображення. Отже, для забезпечення виконання такого роду корекції поставимо задачу реалізації гамма-коректором залежності вигляду

$$E(x,y) = f(E_{\tilde{o}\hat{o}}(\tilde{o}, \hat{o})), \quad (2.90)$$

де f – деяка довільна функція від значень інтенсивностей даної колірної складової в оброблюваному зображенні. Ця функція повинна мати область припустимих значень від 0 до 1, аналогічно функції $E(x,y)$, і реалізовуватися для кожного колірному каналу.

Зазначимо, що в цифровій відеосистемі інтенсивність E , відповідно до вимог, пропонованими до професійної апаратури обробки зображень, повинна мати розрядність не менш 8 біт. При цьому інтенсивність, рівна 0, подається кодом 00000000 у двійковій системі числення, а рівна 1 – кодом 11111111.

Введемо деяку множину функцій $F = \{f_1(E_{ex}), \dots, f_i(E_{ex}), \dots, f_m(E_{ex})\}$ і деякий оператор α вибору функції з множини F . Отже, задача корекції зводиться до апаратної реалізації залежності вигляду:

$$\mathring{A}(x, y) = \alpha F(E_{\tilde{\alpha}\tilde{\delta}}(\tilde{\delta}, \tilde{\delta})). \quad (2.91)$$

Потужність множини F визначається кількістю комбінацій різних типів передавальної і приймальної апаратури, що працює із системою обробки.

Одним з найзручніших способів задання функцій корекції в керуючій програмі є система координат, по осі X якої відкладаються 256 значень інтенсивності колірної складової джерела, а по осі Y – 256 значень інтенсивності колірної складової результату. Як вихідна вибирається функція $E = E_{ex}$, якій відповідає пряма, що йде з початку координат під кутом 45 градусів до осі абсцис (рис. 3.7,а). На цій прямій за допомогою мишки вибираються опорні точки, що можуть переміщатися в площині XOY , змінюючи вид залежності. Плавність переходів від однієї такої точки до іншої може забезпечуватися під час використання апарата кривих Безьє. Такий інтерфейс застосований у відомих графічних редакторах, наприклад, Adobe Photoshop, Corel Draw, що робить простим і «безболісним» перехід монтажників і дизайнерів, ознайомлених з такими редакторами, на нові апаратно-програмні засоби обробки динамічних зображень. На рис. 2.17,б і 2.17,в наведено задані в такий спосіб функції корекції.

Розглянемо один з можливих способів реалізації гамма-корекції динамічних зображень. Важливо враховувати, що така обробка має проводитися в реальному масштабі часу для кожної точки зображення. Так, при стандартній тривалості видимої частини рядка 52 мкс і 768 точок у рядку час виконання корекції (реалізації співвідношення (2.91)) для кожної точки визначається частотою розгорнення растра 14,8 МГц і дорівнює 68 нс. Оскільки функція $f_i(E_{ex})$ має довільний вигляд, найпростішою з апаратної точки зору й водночас найбільш гнучкою є таблична реалізація цієї функції. Таблиця будується за результатами аналізу заданої користувачем залежності і

зберігається в оперативному запам'ятовуючому пристрої (ОЗП). Для збереження однієї такої таблиці достатньо 256 байт пам'яті. Оператор α реалізується шляхом вибору відповідної таблиці з ОЗП.

Як правило, система обробки зображень працює одночасно з декількома різними джерелами. При цьому часто виникає задача мікшування зображень від двох таких джерел. Необхідно врахувати, що в цьому випадку слід робити гамма-корекцію для кожного джерела окремо. Далі розглянемо основи реалізації операції мікшування і покажемо, що мікшуючий блок можна об'єднати з гамма-коректором.

Розглянемо мікшування зображень від двох джерел. Введемо поняття коефіцієнта присутності. Цей коефіцієнт виражає частку інтенсивності колірної складової в даній точці результуючого зображення, взяту від інтенсивності цієї колірної складової в тій же точці вихідного зображення. Введемо також деяку функцію присутності $K=K(x,y)$, що виражає залежність значення коефіцієнта присутності K від координат (x,y) точки на екрані. Тоді операцію мікшування можна аналітично подати у вигляді

$$\begin{aligned} E_R(x, y) &= K_{R1}(x, y)E_{R1}(x, y) + K_{R2}(x, y)E_{R2}(x, y); \\ E_G(x, y) &= K_{G1}(x, y)E_{G1}(x, y) + K_{G2}(x, y)E_{G2}(x, y); \\ E_B(x, y) &= K_{B1}(x, y)E_{B1}(x, y) + K_{B2}(x, y)E_{B2}(x, y), \end{aligned} \quad (2.92)$$

де буквенний індекс визначає приналежність до колірної складової, а цифровий індекс – до джерела зображення.

У виразах (2.92) функція присутності різна для різних колірних складових.

Як видно з виразів (2.92), мікшування відбувається однаково в різних колірних каналах. Тому далі, для спрощення аналізу, вестимемо обговорення для довільного колірної складової. Тоді кожний з виразів (2.92) можна подати у вигляді

$$E(x, y) = K_1(x, y)E_1(x, y) + K_2(x, y)E_2(x, y). \quad (2.93)$$

З'ясуємо залежність між функціями присутності K_1 і K_2 . Як уже було відзначено $E(x, y)$, $E_1(x, y)$, $E_2(x, y)$ змінюються в інтервалі від 0 до 1. Звідси випливає, що

$$K_1(x, y) + K_2(x, y) \leq 1. \quad (2.94)$$

Якщо умова (2.94) не виконується для деякої точки, то в цій точці можливе спостереження насичення даної колірної складової, якщо права частина виразу (2.93) буде більше 1. У такій точці відбудеться втрата інформації про зображення, що змішуються. Слід також зазначити, що функція присутності не може бути негативною, оскільки в цьому випадку можливе одержання негативної інтенсивності, що не відповідатиме області припустимих значень функції E . Тоді обмеження на функції K_1 і K_2 приймуть вигляд:

$$\begin{aligned} 0 &\leq K_1(x, y) \leq 1; \\ 0 &\leq K_2(x, y) \leq 1; \\ K_1(x, y) + K_2(x, y) &\leq 1. \end{aligned} \quad (2.95)$$

З дотриманням умов (2.95) втрата інформації про колірну складову у даній точці зображення не відбудеться ні при яких значеннях інтенсивностей цієї колірної складової в тих же точках зображень, що мікшуються.

2.3.2 Базова архітектура колориметричного обчислювача

Загальна структура одного каналу пристрою, запропонованого в попередньому розділі, наведена на рис.2.25.

Як випливає з рисунка, значення функцій E і K надходять на блоки, що множать – блок $M1$ для зображення від джерела 1 і блок $M2$ для зображення від джерела 2. Результати перемножувань складаються суматором SM , з виходу

якого знімається значення інтенсивності даного кольору для результуючого зображення. Необхідно розуміти, що при реалізації цифрового мікшера така структура призведе до істотних апаратних витрат. Це пов'язано з тим, що інтенсивність E і функція присутності K , відповідно до вимог до професійної апаратури обробки зображень, повинні мати розрядність не менш 8 і 6 біт відповідно. У такому випадку кожен блок, що перемножує, має обробляти 8-ми і 6-бітні значення і видавати 8-бітний результат. Як результат при цьому вибираються старші 8 біт добутку таких кодів. Крім того, результат мікшування, а отже, і результат обох перемножувань, має формуватися з частотою відображення точок на екрані при розгорненні зображення. При 768 точках у рядку ця частота дорівнює 14,8 МГц (період, відповідно, 68 нс).

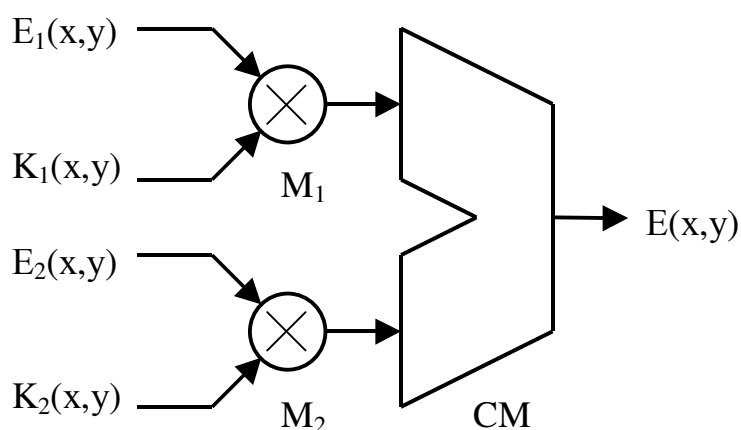


Рисунок 2.25 – Загальна структура одного каналу блока, що мікшує

Розглянемо структуру, наведену на рис.2.26.

Як видно з рисунка, структура містить один блок M , що перемножує, який дозволяє одержати два добутки за один період тактової частоти. На входи цього блока протягом першого напівперіоду тактової частоти, за рахунок застосування елементів, що мультиплексують, надходять інтенсивність кольору і значення функції присутності для зображення від джерела 1, а протягом 2-го напівперіоду – для зображення від джерела 2. У такій структурі між блоком, що перемножує, і суматором застосовані додаткові регістри для збереження

результатів двох перемножень. На наступному такті ці значення складаються суматором. Отже, структура припускає наявність двотактного конвеєра, що дозволяє одержувати результат мікшування з необхідною частотою, але з затримкою на 2 такти. На рис. 2.26 більш детально наведено вхідні блоки, що є регістрами, для збереження оброблюваних у даний момент значень інтенсивностей і функцій присутності.

Нехай операція мікшування виконується разом з гамма-корекцією кожного з джерел, що мікшуються. Тоді співвідношення (2.93) з урахуванням (2.91) прийме вигляд:

$$E(x, y) = K_1(x, y)\alpha_1 F(E_1(x, y)) + K_2(x, y)\alpha_2 F(E_2(x, y)). \quad (2.96)$$

У залежності (2.16) оператор α_1 визначає, яка функція корекції з множини функцій F вибирається для гамма-корекції джерела 1, а оператор α_2 – аналогічно для джерела 2.

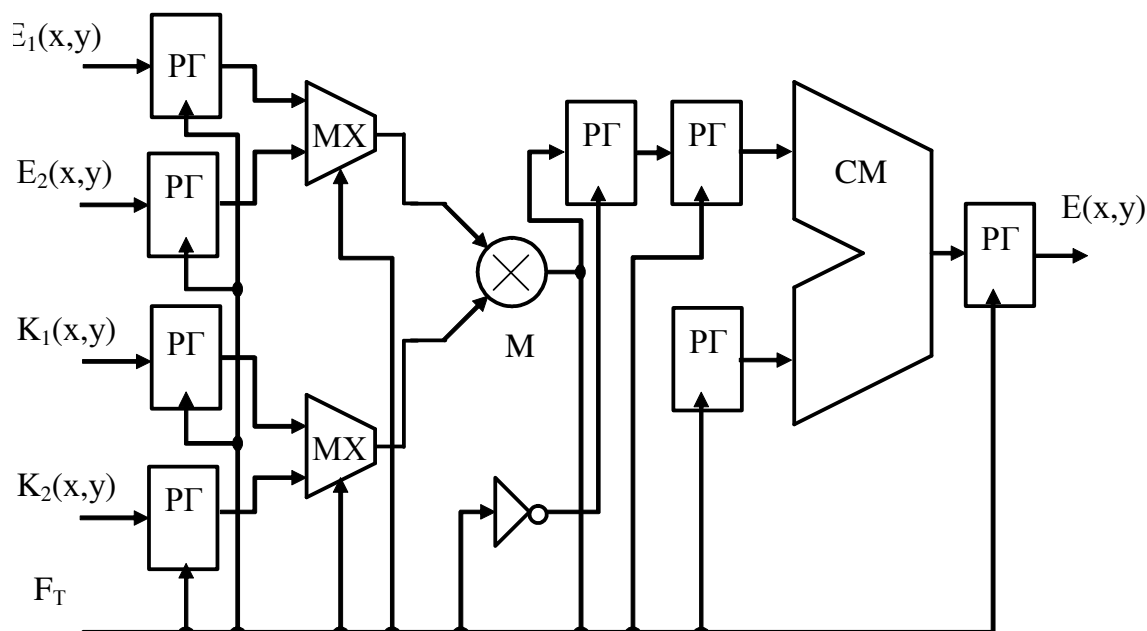


Рисунок 2.26 – Структура блоку, що мікшує, на основі одного множника

Внесемо функцію корекції під оператор α . У цьому випадку множину F функцій корекції необхідно доповнити функціями, що є добутками базових функцій корекції на всі можливі значення функції присутності. У цифровій системі кількість таких значень обмежена і визначається кількістю розрядів, необхідних для подання функції присутності. За наявності 4-х базових функцій корекції і під час використання 6-розрядної функції присутності множина F має включати 256 різних функцій, кожна з якої, при 8-бітному поданні інтенсивності, має 256 значень. Отже, множина F може зберігатися в оперативному запам'ятовуючому пристрої ємністю 64 кілобайта (256 функцій по 256 одnobайтових значень). Основною вимогою до такого ОЗП стосовно до структури, наведеної на рис.2.26, має бути забезпечення можливості двох вибірок протягом періоду тактової частоти розгорнення. Серед представленої на вітчизняному ринку імпортової елементної бази присутні ІМС статичного ОЗП ємністю 64 кілобайта і вибірки, що мають час, 15 нс, достатньо для виконання вищевказаної вимоги.

Отже, у колірному каналі в основу гамма-коректора і блока, що перемножує, пристрою, що мікшує, покладено ОЗП ємністю 64 кілобайта. Структура такого блока наведена на рис.2.27.

У такій структурі функція присутності й оператор вибору функції корекції об'єднані в керуючий α -канал, що є додатковими інформаційними бітами, аналіз яких визначає, які саме перетворення виконуватимуться для даної точки. При цьому біти, що відповідають за вибір базової функції корекції, постійні для даного типу джерела, а біти, що визначають значення функції присутності, можуть змінюватися для кожної точки зображення, що визначається виглядом функції $K(x,y)$. Слід зазначити, що α -канал формується окремо для кожного з двох зображень, що мікшуються, і в кожному колірному каналі. Такий канал можна описати виразом:

$$E(x, y) = \begin{cases} 0, & \alpha_1 F(E_1(x, y)) + \alpha_2 F(E_2(x, y)) < 0 \\ \alpha_1 F(E_1(x, y)) + \alpha_2 F(E_2(x, y)), & 0 \leq \alpha_1 F(E_1(x, y)) + \alpha_2 F(E_2(x, y)) \leq 1. \\ 1, & \alpha_1 F(E_1(x, y)) + \alpha_2 F(E_2(x, y)) > 1 \end{cases} \quad (2.97)$$

У виразі (2.97) передбачена перевірка входження значення інтенсивності, що обчислюється, у рамки області припустимих значень інтенсивності. Якщо отримане значення виходить за рамки цієї області, що коректує блок як значення інтенсивності підставляє значення відповідної межі області припустимих значень.

Розглянемо принцип дії пристрою, побудованого відповідно до структури, наведеної на рис.2.26. Протягом першого напівперіоду тактової частоти в кожному з колірних каналів значення інтенсивності відповідної колірної складової і значення з α -каналу надходять на адресні входи ІМС ОЗП. Значення функції, що зчитується, зберігається в регістрі для подальшої обробки. Протягом другого напівперіоду аналогічно одержуємо значення функції для зображення від другого джерела. Далі, протягом наступного періоду тактової частоти, значення, отримані на попередньому такті, надходять на АЛП, кероване додатковим α -каналом. Операцію додавання можна замінити однією з функцій, додатково реалізованих ІМС АЛП (операнд 1, операнд 2, інверсія операнда 1 чи 2, бінарні логічні операції).

Пристрій у цілому має три ідентичних канали, що мають конвеєрну структуру. Конвеєр має два такти і забезпечує одержання результату з частотою 14,8 МГц при 768 точках у рядку. Крім того, є блок керування вибором функцій корекції (БФК) і блок обчислення функцій присутності (БФП), що формують α -канали, блок зв'язку (БЗ) з ЕОМ. Є блок ініціалізації (БІ) ОЗП, що дозволяє ініціалізувати кожен комірку ОЗП окремо. Є також канал обходу блоку корекції (на рис.2.27 не показаний), що дозволяє ініціалізувати ОЗП у будь-який момент часу зі збереженням працездатності

пристрою (при цьому корекція і мікшування не виробляються). Узгодження цих блоків, загальних для всіх трьох каналів, здійснюється блоком керування.

Розглянемо можливі способи задання й обчислення функцій присутності $K(x,y)$. Аналітично ця функція може подаватися у вигляді деякої залежності від координат (x,y) , виведеної в даний момент розгорнення точки зображення. Графічно таку залежність можна подати в тривимірній системі координат $OXYK$ у вигляді деякої поверхні, а також у вигляді проєкцій на площині OJK , OYK і OXY . Якщо розглянута поверхня є поверхнею обертання, її зручно подавати в циліндричній чи сферичній системах координат. Слід враховувати, що функція присутності може бути різною для кожної колірної складової. Для виконання співвідношення (3.15) функції K_1 і K_2 можуть зв'язуватися залежністю:

$$K_2(x, y) = \overline{K_1(x, y)}, \quad (2.98)$$

яка легко реалізується в цифровій системі обробки. Обчислення функцій присутності в залежності від координат (x,y) покладається на відеопроцесор. Принцип його дії такий. Зображення від мікшувальних джерел надходять до блока, що мікшує. За допомогою команд управління визначається, яке зображення обирається як 1-ше, яке – 2-ге.

Крім того, одне з цих зображень можна занести до масочної пам'яті процесора. До цієї пам'яті також можна занести зображення від ЕОМ. Зображення, що знаходиться у масочній пам'яті (масочне зображення), використовується як функція присутності. У цьому разі функція присутності у точці (X,Y) визначається як яскравість точки з відповідними координатами, що знаходиться у масочній пам'яті. У складному випадку функція присутності визначається для кожного каналу кольору як інтенсивність кольорових складаючих у відповідній точці масочного зображення. Читання масочної пам'яті відбувається під управлінням проєктивно-масштабуючого блока. Отже, здійснюється динамічна зміна функції присутності у реальному часі.

Розроблено програмне забезпечення, що дозволяє розроблювати різноманітні масочні зображення та, відповідно, отримувати велику кількість різних засобів мікшування. Існує можливість використання постійного значення функції присутності для усіх точок у межах напівкадру. У такому разі, нове значення функції присутності встановлюється під час кадрового імпульсу гасіння.

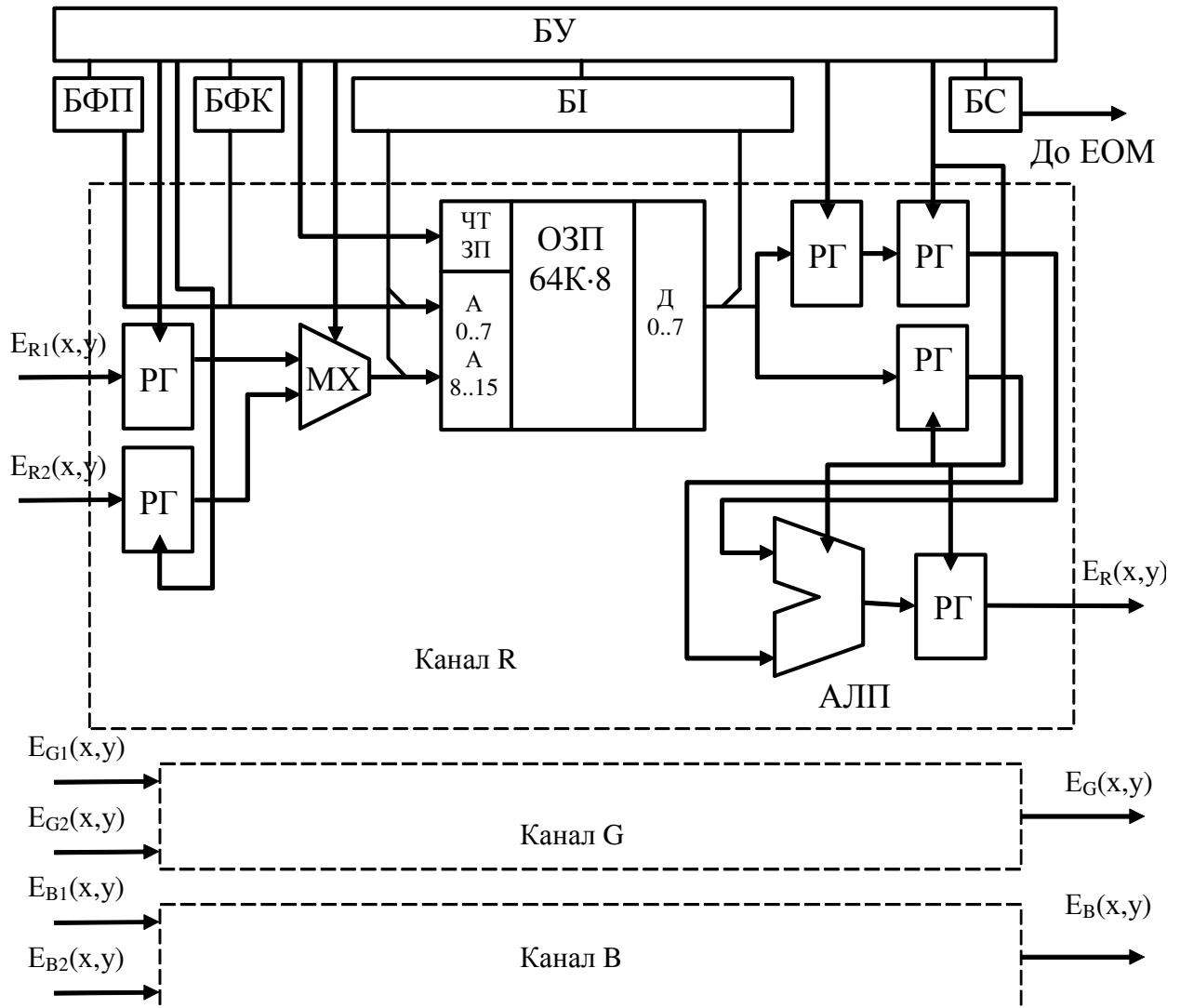


Рисунок 2.27 – Структура блока гамма-корекції і мікшування на основі оперативного запам'ятовуючого пристрою

Демонстрацією можливостей розробленого пристрою є нижче подана підбірка ілюстрацій з поясненням отриманого ефекту (рис. 2.28–2.33).



Рисунок 2.28 – Плавний перехід від кольорового фону до зображення. Функція присутності має вигляд піраміди з чотирма боковими гранями та верхівкою у центрі кадру



Рисунок 2.29 – Ефект витискання. Здійснюється шляхом динамічного мікшування двох зображень. Функція присутності має вигляд зрізаного конусу



Рисунок 2.30 – Проявлення одного зображення крізь інше



Рисунок 2.31 – За допомогою мікшувального блока можливе отримання незвичайних ефектів



Рисунок 2.32 – Мікшування двох зображень. Функція присутності визначається яскравістю одного з цих зображень, занесеного до маскової пам'яті



Рисунок 2.33 – Під час роботи у реальному часі мікшувальний блок дозволяє здійснити плавний перехід від одного зображення до іншого. Нове зображення поширюється з центру старого

2.3.3. Колориметричний коректор зображень

Колірна корекція має виконуватися для кожного вхідного джерела незалежно від інших джерел. Структуру колірною коректора, в основі якого лежить апроксимація передавальної характеристики мережею СМАС, наведено на рис. 2.34. У її основі лежить мережа СМАС, що має три вхідні змінні і що обчислює три функції.

Для реалізації мережі є три ідентичні блоки кодування змінних R, G, B (відповідно БКЗ R, БКЗ G, БКЗ B), ОЗП для зберігання добуток базисних функцій (ОЗП ДБФ), загальне для трьох функцій, що реалізуються, ОЗП для зберігання ваг для кожної колірної складової (ОЗП ваг R, G і B), помножувачі M1 і накопичуючі суматори $\Sigma 1$, обчислюючи добутки векторів ваг і коефіцієнтів. У даній структурі використовується колориметрична система RGB, як найбільш наочна і така, що забезпечує якнайкращу якість зображення. Для системи YUV структура буде ідентичною. Зміниться лише інтерфейс задання передавальної характеристики. Подібно до розглянутої в структурі мікшуючого блока є елементи (мультиплексори MX1, MX2R, MX2G, MX2B і буферні елементи B1, B2R, B2G, B2B) для ініціалізації використовуваних у структурі ОЗП значеннями від управляючої ЕОМ.

У розглянутій структурі є елемент «логічне І», який виконує кероване маскування старших розрядів формованої адреси в ОЗП ваг. Хешування застосоване для скорочення об'єму передаваних з ЕОМ у відеосистему даних. Так, при заданні передавальної характеристики оператор бажає бачити зміни на реальному відеопотоці. Для цього вводиться великий коефіцієнт хешування, чим забезпечується скорочення об'єму ваг, які настроюються, і передавальних. При цьому якість апроксимації буде понижена, але достатня для візуальної оцінки виконуваних коректувань. Коли буде отриманий необхідний результат, ступінь хешування зменшується або взагалі не використовується. Настроюються і передаються всі ваги, а якість апроксимації буде максимальною.

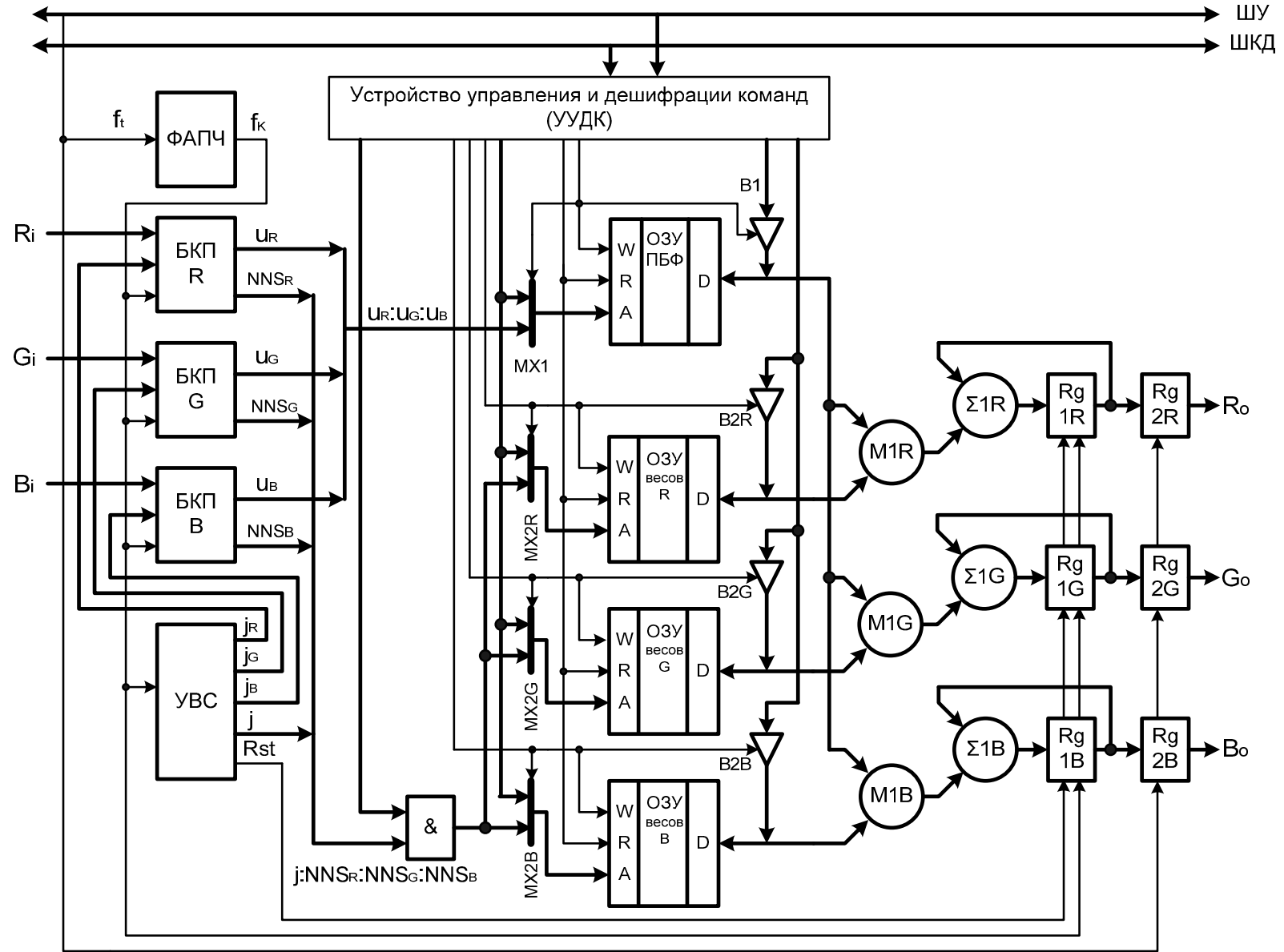


Рисунок 2.34 – Структура колірного коректора

На відміну від синтезатора функцій присутності, вхідні змінні колірною коректора змінюються непослідовно. Внаслідок цього пропонується використовувати блоки кодування змінних, що працюють із значеннями ρ , рівними степеню числа 2. Відмінність в тому, що на виході БКЗ – не значення базисної функції, а її аргумент u . Ці аргументи шляхом конкатенації розрядів об'єднуються і поступають на адресний вхід ОЗП зберігання добутоків базисних функцій. Основний ефект від такої реалізації полягає в скороченні апаратних і тимчасових витрат на реалізацію обчислення добутку трьох змінних і спрощенні схем ініціалізації ОЗП. При значеннях $\rho = 4, 8, 16$ і 32 об'єм ОЗП ДБФ складе відповідно $64, 512, 4096$ і 32768 комірок.

Система ФАПЧ загальна для всіх трьох БКЗ і призначена для синтезу частоти кодування.

Пристрій вибірки ступеня формує номери ступенів кодування для кожного БКЗ. Воно може реалізуватися як лічильник з непослідовним рахунком, або як лінійний рекурентний регістр з доданням комбінації «всі нулі».

Головними перевагами пропонованої реалізації є скорочення об'єму пам'яті і переданих даних в порівнянні з табличним методом при збереженні універсальності і широких можливостей колірної корекції, які забезпечуються за рахунок синтезу передавальних характеристик, залежних від усіх колірних складових зображення. Для оцінки скорочення об'ємів пам'яті наведемо числові значення, зведені в таблицю 2.4.

У таблиці наведено об'єми пам'яті для 8-бітового подання колірних складових. Для пропонованої моделі об'єми вказані без виконання хешування. Як видно з наведених даних, економія об'ємів складає від $15,5$ до 726 разів. Із застосуванням хешування ці показники можуть збільшуватися ще в кілька разів.

Таблиця 2.4 – Об'єми пам'яті, необхідні для реалізації колірної корекції

Табличний метод колірної корекції, комірок ОЗП	Пропонований метод колірної корекції, комірок ОЗП			
	$\rho = 4$	$\rho = 8$	$\rho = 16$	$\rho = 32$
3·16777216	3·1085874	3·284242	3·77744	3·23086

2.3.4. Синтез функцій присутності при мікшуванні

Мікшуючий блок для роботи з відеосигналами в реальному часі, що використовує для синтезу функцій присутності мережу СМАС, наведено на рис. 2.35.

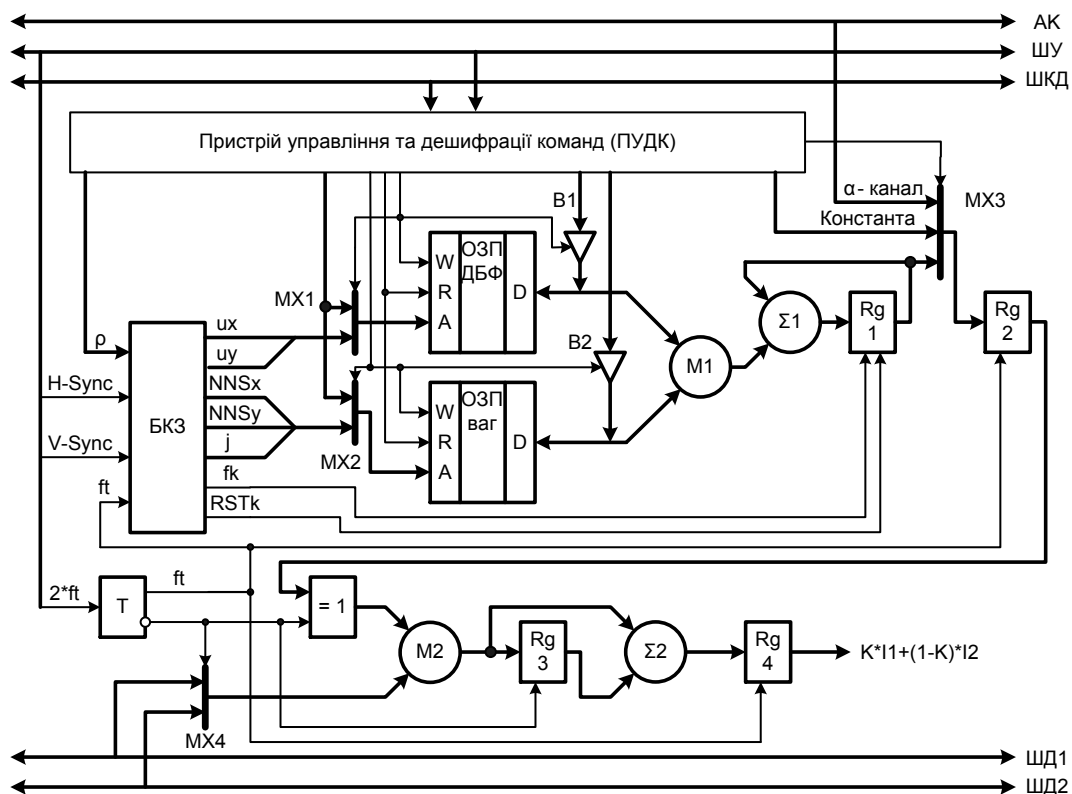


Рисунок 2.35 – Структура мікшуючого блока

Структуру мережі СМАС утворюють блок кодування змінних БКЗ, оперативні запам'ятовуючі пристрої для зберігання значень добутків базисних функцій (ОЗП ДБФ) і значень ваг (ОЗП ваг), помножувач M1 і накопичуючий суматор $\Sigma 1 - Rg1$. Мультиплекси MX1 і MX2, а також буферні елементи B1 і B2 необхідні для ініціалізації ОЗП. БКЗ як вхідні змінні використовує сигнали

тактової частоти f_t і рядкової синхронізації H-Sync. Ці сигнали відповідають послідовно перебираним координатам x і y поточної точки в мікшованих зображеннях.

Обчислене мережею СМАС значення функції присутності для даної точки через мультиплексор МХ3 поступає на регістр Rg2, в якому зберігається протягом тривалості точки. Мультиплексор МХ3 призначений для забезпечення можливості мікшування зображень з використанням постійного для всього кадру значення функції присутності, або з використанням α -каналу. Останній зручний для виведення на зображення логотипів або титрів різного ступеня прозорості і складності.

Операція мікшування виконується на помножувачі М2 і суматорі $\Sigma 2$. Першу половину періоду тактової частоти через мультиплексор МХ4 на помножувач М2 поступає значення інтенсивності колірною складовою з шини даних ШД1. На другий вхід помножувача через керований інвертор $=1$ (за своєю суттю це схема що "Виключне АБО", на один вхід якої подано значення функції присутності, а на другий – тактова частота проходження точок) поступає значення функції присутності в прямому коді. Після закінчення першої половини періоду обчислене помножувачем значення твору $K \cdot I_1$ захищується в регістрі Rg3. Протягом другої половини періоду тактової частоти на помножувач М2 поступають значення інтенсивності з шини даних ШД2 і значення функції присутності в зворотному коді. Обчислюється добуток $(1-K) \cdot I_2$ і поступає на суматор $\Sigma 2$, на який також поступає збережене в регістрі Rg3 значення добутку $K \cdot I_1$. Обчислене значення суми добутків захищується в регістрі Rg4.

Теоретично область допустимих значень функції присутності визначається інтервалом $[0,1]$. Проте на практиці використання дробових значень призводить до ускладнення обчислень і втрати їх наочності. Тому фактично пропонується вважати областю визначення функції присутності інтервал $K=[0,255]$, а результуюче значення інтенсивності визначати із співвідношення:

$$I(x, y) = \frac{K(x, y)I_1(x, y)}{256} + \frac{(255 - K(x, y))I_2(x, y)}{256}, \quad K(x, y) \in [0, 255]. \quad (2.99)$$

Внаслідок того, що операція ділення на 256 виконується шляхом зрушення результату множення на 8 розрядів вправо, або що те ж саме, шляхом узяття як результат перемножування старших 8-ми розрядів результату.

Для заміни виразу $(I-K)$ на $(255-K)$ в (3.19) зробимо таке припущення: за 1 приймається значення $K=255$, а не 256, як цього слід було чекати. Таке припущення не призводить до видимих змін в результуючому зображенні і може виконуватися шляхом інвертування значення K . Це дозволяє дещо скоротити апаратні витрати на переклад зворотного коду в додатковий.

В цілому реалізація ефектів мікшування з використанням структури, наведеної на рис. 2.12, полягає в такому. Оператор, що працює за управляючою ЕО, створює зображення шторки або фігури шляхом її побудови в будь-якому графічному редакторі. Можлива також генерація шторки або фігури шляхом програмної реалізації деякої аналітичної залежності. Сформоване зображення фігури або шторки подається програмній моделі мережі СМАС, яка обчислює значення ваг, необхідні для апроксимації цього зображення. Отриманий масив ваг передається інтерфейсом ЕОМ – блок зв'язку відеопроцесора. Далі адреси і значення ваг по шині команд і даних ШКД поступають на мікшуючий блок, де після дешифрування команди в блоці ПУДК формуються відповідні адреси і дані для ініціалізації ОЗП ваг. Після ініціалізації ОЗП ваг мікшуючий блок готовий до роботи.

Передбачена зміна значення ρ і виду базисних функцій забезпечує можливість змінювати співвідношення якість/швидкість ефекту мікшування. Так, при великих значеннях ρ погіршується якість апроксимації, але зменшується об'єм передавальних параметрів. Якість апроксимації залежить також і від того, який вид базисних функцій використовується для тієї або іншої функції присутності.

Головною перевагою пропонованої моделі задання функцій присутності, в порівнянні з табличною, є менший об'єм передавальних параметрів, а в порівнянні з алгоритмічною – висока універсальність, тобто застосовність для несиметричних функцій присутності довільного вигляду. Порівняльний аналіз об'ємів передавальних даних у разі табличного і запропонованого способів синтезу функцій присутності для зображень розміром 720 на 288 точок (стандартний телевізійний напівкадр) наведено в таблиці 2.5.

Таблиця 2.5 – Об'єми пам'яті, необхідні для синтезу функцій присутності

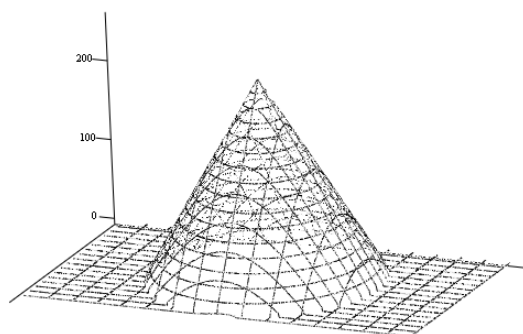
Табличний метод для фігур		Пропонований метод для довільних фігур			
довільних	симетричних	$\rho = 4$	$\rho = 8$	$\rho = 16$	$\rho = 32$
207360 ос.	51840 ос.	52600 ос.	26810 ос.	13920 ос.	7488 ос.

З таблиці 2.4 видно, що при $\rho = 4$ пропонована модель для довільних фігур практично не поступається табличною для симетричних відносно обох осей фігур і майже в 4 рази скорочує об'єм пам'яті для довільних фігур. Вже при $\rho=8$ пропонована модель майже в 2 рази скорочує об'єм пам'яті навіть в порівнянні з табличною для симетричних фігур і майже в 8 разів для довільних. Великі значення ρ дозволяють скоротити об'єм пам'яті від 4 до 27 разів.

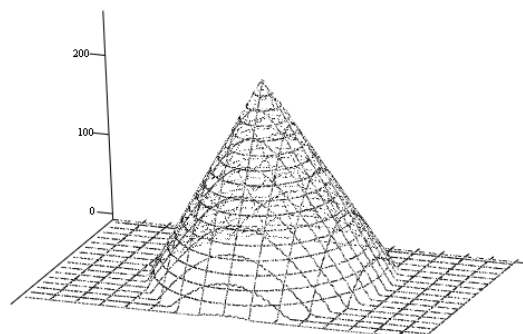
У структурі на рис. 2.35 не використовується хешування вектора асоціацій. Це пов'язано з тим, що необхідні об'єми пам'яті достатньо малі, а виникнення хеш-колізії може призвести до небажаних артефактів на функції присутності, що не виправдовується ще більшим скороченням об'ємів пам'яті.

Нижче наведено приклади апроксимації функцій присутності кінцевого (рис. 2.36) і сферичного (рис. 2.37) вигляду. На рис. 2.36, а і рис. 2.37, а наведено функції, отримані за аналітичними залежностями. На рис. 2.36, б, в, г і рис. 2.37, б, в, г – функції, отримувані в результаті навчання мережі з базисними функціями у вигляді сплайнів першого порядку і значеннями параметра ρ , рівними 8, 16, 32 відповідно. На рисунках видно, що із збільшенням значення ρ відбувається погіршення апроксимуючих властивостей

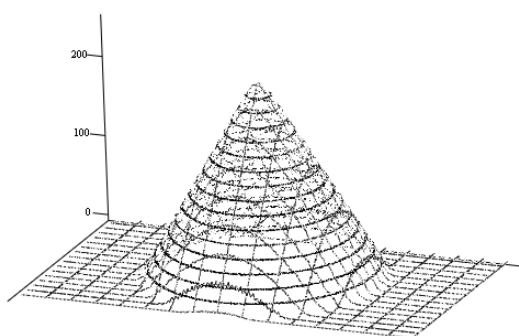
мережі, відбувається згладжування різких переходів і з'являється шум. Проте, як видно з рис. 2.36, д, е і рис. 2.37, д, е, на яких наведено результати мікшування зображень, виконаної з використанням функцій присутності, обчислених аналітично і мережею СМАС з параметром $\rho = 32$ відповідно, погіршення якості апроксимації непомітне на результуючому зображенні.



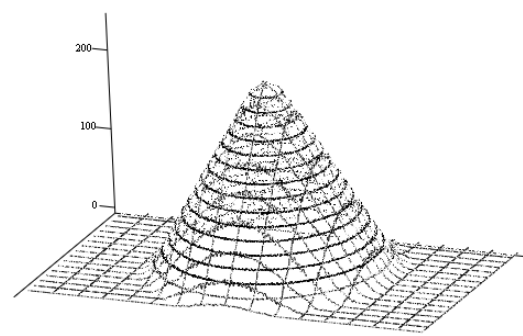
а



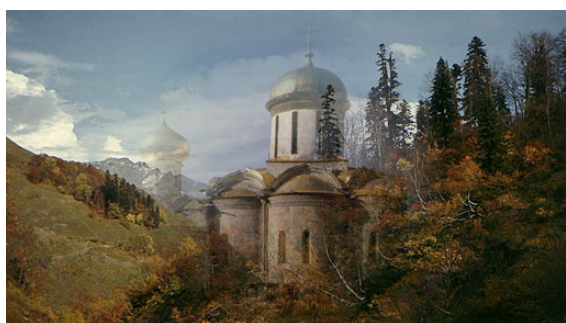
б



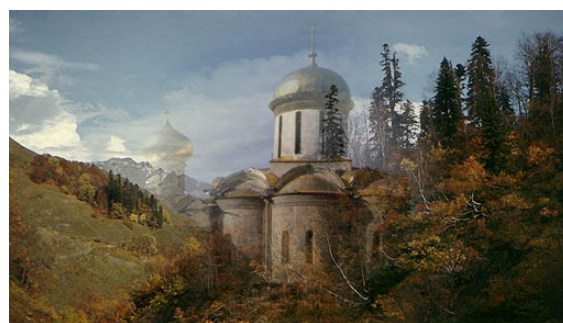
в



г

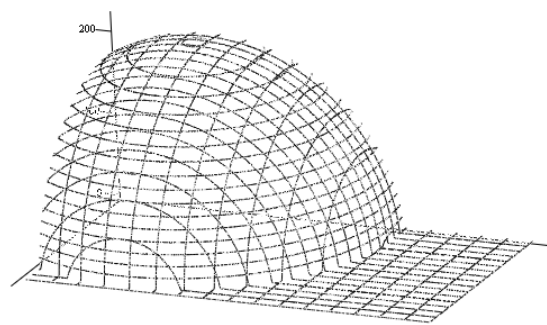


д

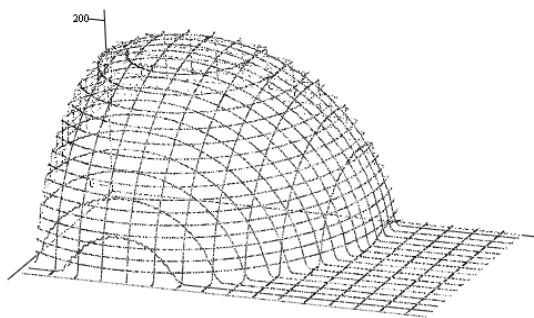


е

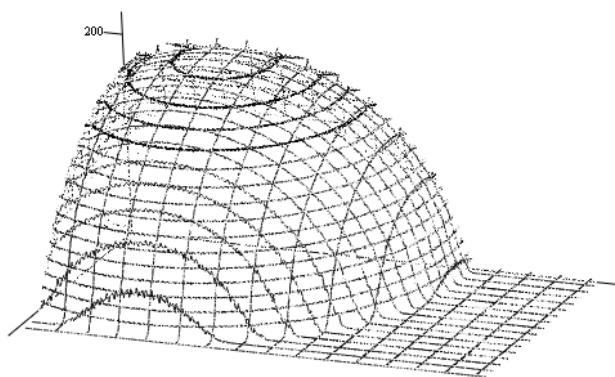
Рисунок 2.36 – Функція присутності кінчного вигляду



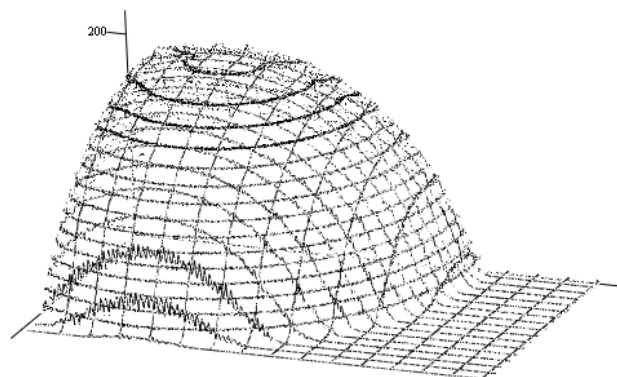
а



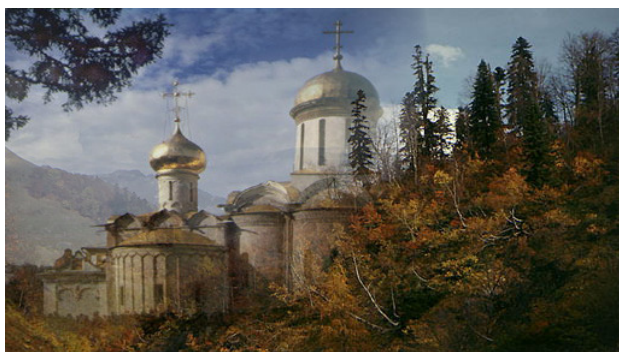
б



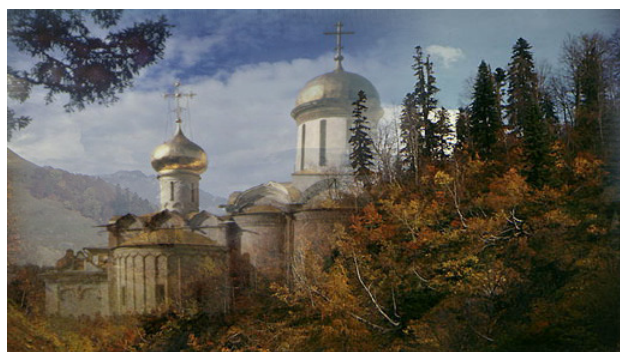
в



г

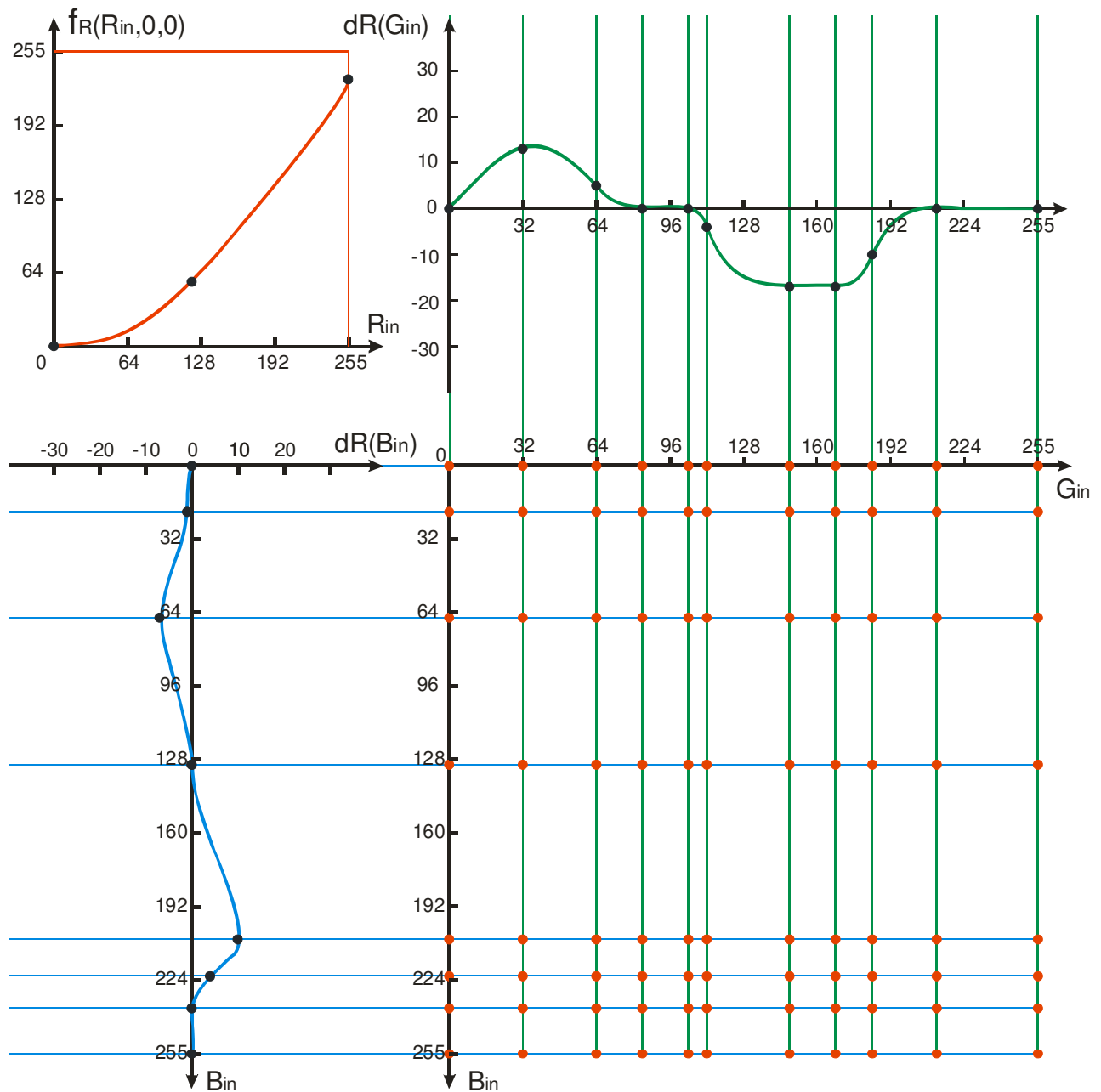


д



е

Рисунок 2.37 – Функція присутності сферичного вигляду



Кількість опорних точок залежності $dR(G_{in})$: $N_{R(G)} = 11$;

Кількість опорних точок залежності $dR(B_{in})$: $N_{R(B)} = 8$;

Кількість опорних точок залежності $f_R(R_{in}, 0, 0)$: $N_{f(R)} = 3$;

Загальна кількість навчальних точок:

$$N = N_{R(G)} \cdot N_{R(B)} \cdot N_{f(R)} = 11 \cdot 8 \cdot 3 = 264;$$

Значення передавальної функції R_{out} у точці (R_i, G_j, B_k) :

$$R_{out}(R_i, G_j, B_k) = f_R(R_i, 0, 0) + dR(G_j) + dR(B_k)$$

Рисунок 2.38 – Завдання опорних точок колірної корекції

2.3.5. Реалізація структури синтезаторів функцій присутності

Модель блока кодування змінних, що послідовно змінюються, наведено на рис.2.36. У моделі вхідні змінні явно не задаються. Як координата x точки на екрані виступає частота кодування Fk , рівність нулю змінної x визначається за початком рамки рядка, що формується з сигналу рядкової синхронізації $HSync$. Сигнал $HSync$ також використовується як координата y , черговому значенню якої відповідає початок нового рядка. Нульове значення змінної y визначається по початку рамки кадру, формованій з сигналу кадрової синхронізації $VSync$. Тактування блока здійснюється частотою Fk , яка формується системою ФАПЧ (на структурі не показано) з подвоєної частоти Clk виведення точок зображення. На виході блок кодування змінних формує значення аргументів базисних функцій u_x і u_y для змінних x і y відповідно, а також номери активних на даному ступені кодування нейронів NNS_x і NNS_y , значення номера ступеня кодування j і сигнал Nxt_Val почала кодування чергових значень змінних.

Модель мікшуючого блока, в якому для синтезу функції присутності використовується нейронна мережа СМАС, наведено на рис. 2.40. Колірні складові R , G і B мікшуваних зображень поступають шинами даних $BD1$ і $BD2$, захищуються у вхідних регістрах і через визначаючий мультиплексор, частка присутності якого із зображень обчислюється зараз, поступають на помножувач. На другий вхід помножувача поступає пряме або інверсне значення (формоване за допомогою елементів що "ВИКЛЮЧНЕ АБО") функції присутності K . Значення добутку захищується в регістрі, звідки поступає на накопичуючий суматор, що обчислює суму часток присутності кожного із зображень. Значення суми захищується у вихідному регістрі і є результатом мікшування. Сигнали управління процесом мікшування формуються з тактової частоти Clk .

Значення функції присутності K за допомогою мультиплексора обирається або з альфа-каналу AK , або сформоване синтезатором. На основі сигналів тактових частот, рядкової і кадрової синхронізації блок кодування змінних *Var_Coding_Block* формує аргументи базисних функцій, які поступають як адреса на ОЗП добутоків базисних функцій, реалізованого всередині ПЛІС, а також номери активних нейронів і ступеня кодування, які після конкатенації поступають на шину адреси зовнішнього ОЗП ваг.

Вибрані з ОЗП вагів значення поступають на помножувач, що обчислює їх добуток із значенням твору базисних функцій, а потім на накопичуючий суматор, що обчислює значення функції. Ініціалізація внутрішнього і зовнішнього ОЗП здійснюється по шині команд і даних *BCD* під управлінням стробуючих сигналів, що формуються блоком управління системою обробки.

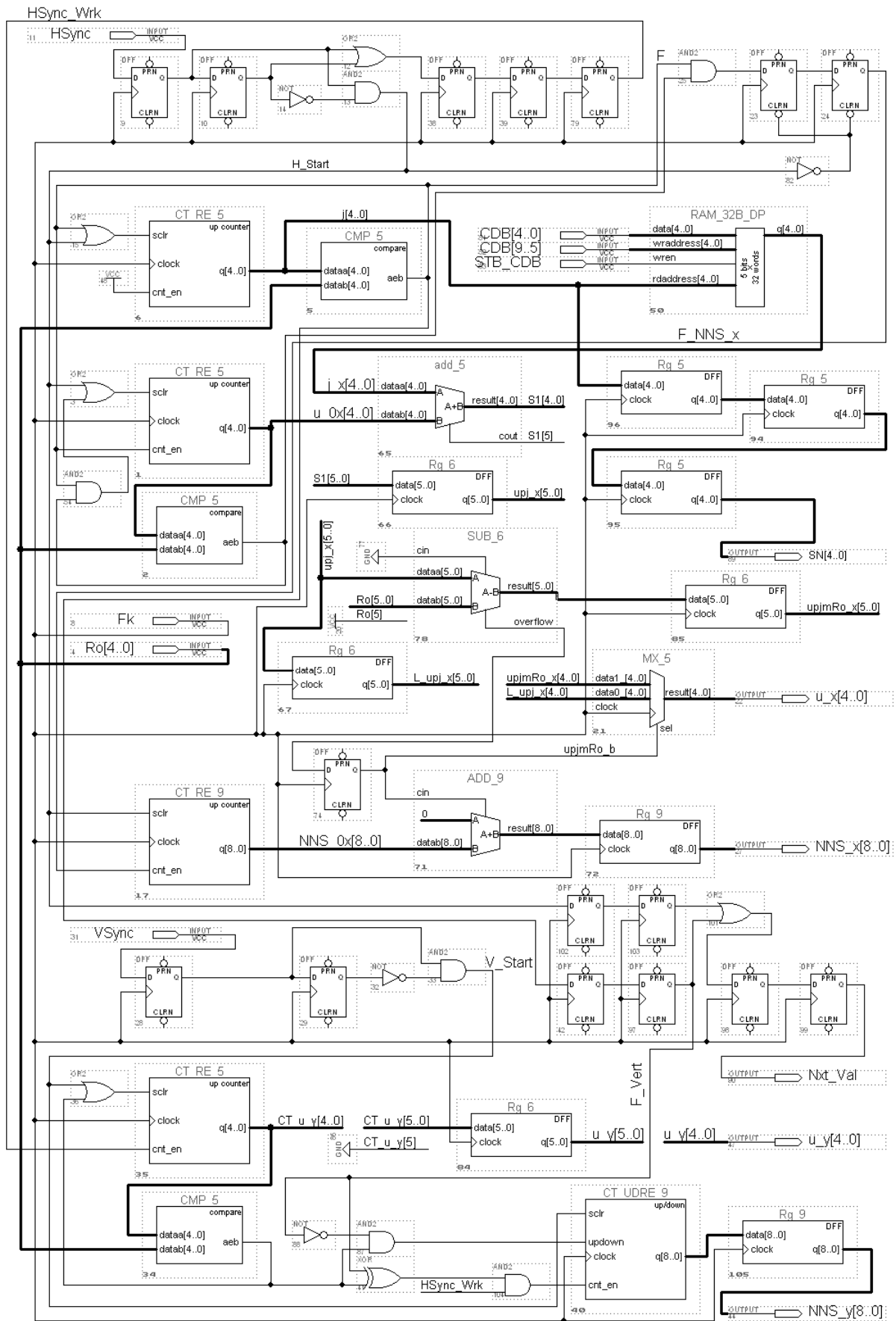


Рисунок 2.39 – Блок кодування змінних, що змінюються послідовно

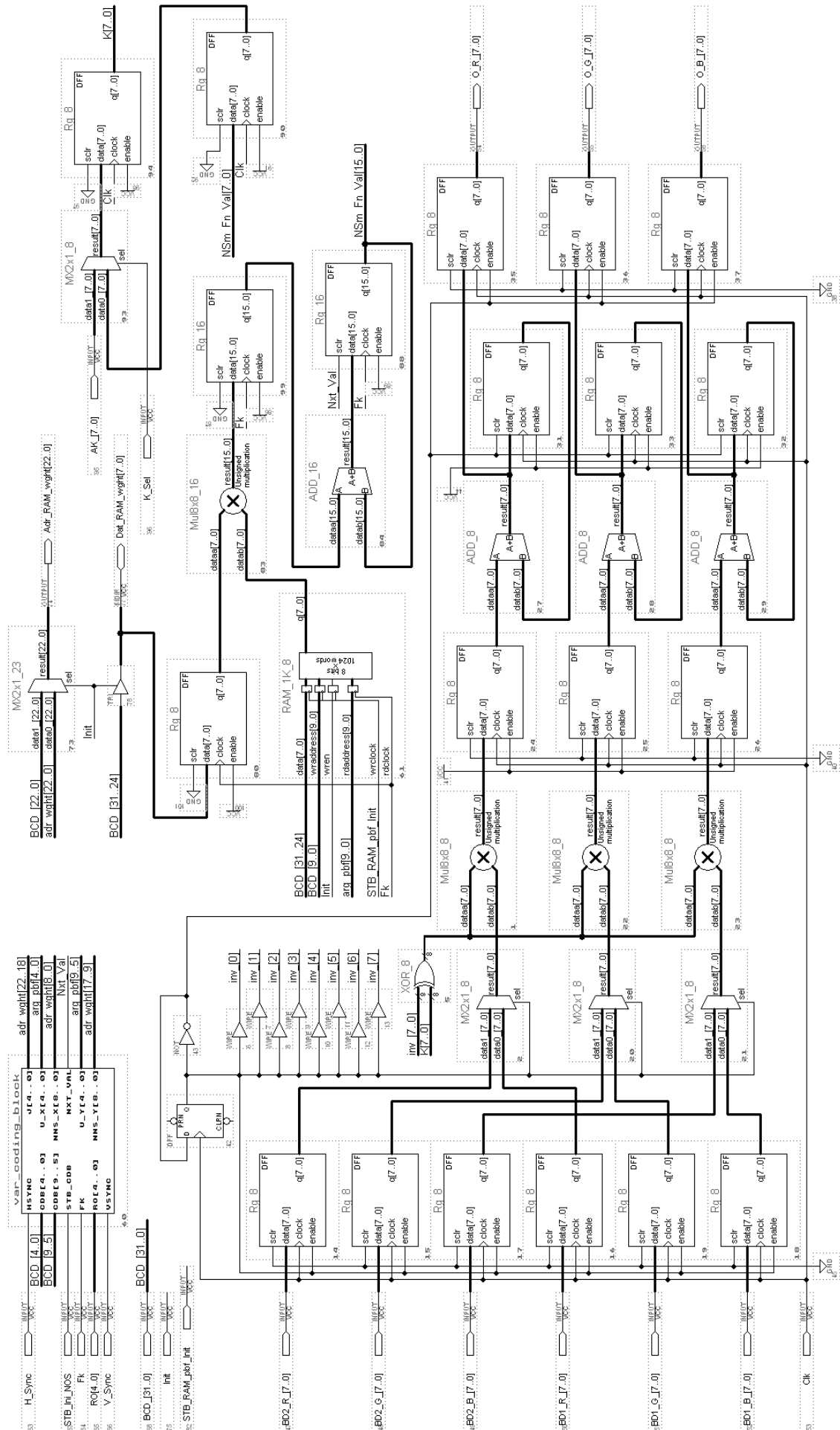


Рисунок 2.40 – Мікшуючий блок на основі нейронної мережі СМА

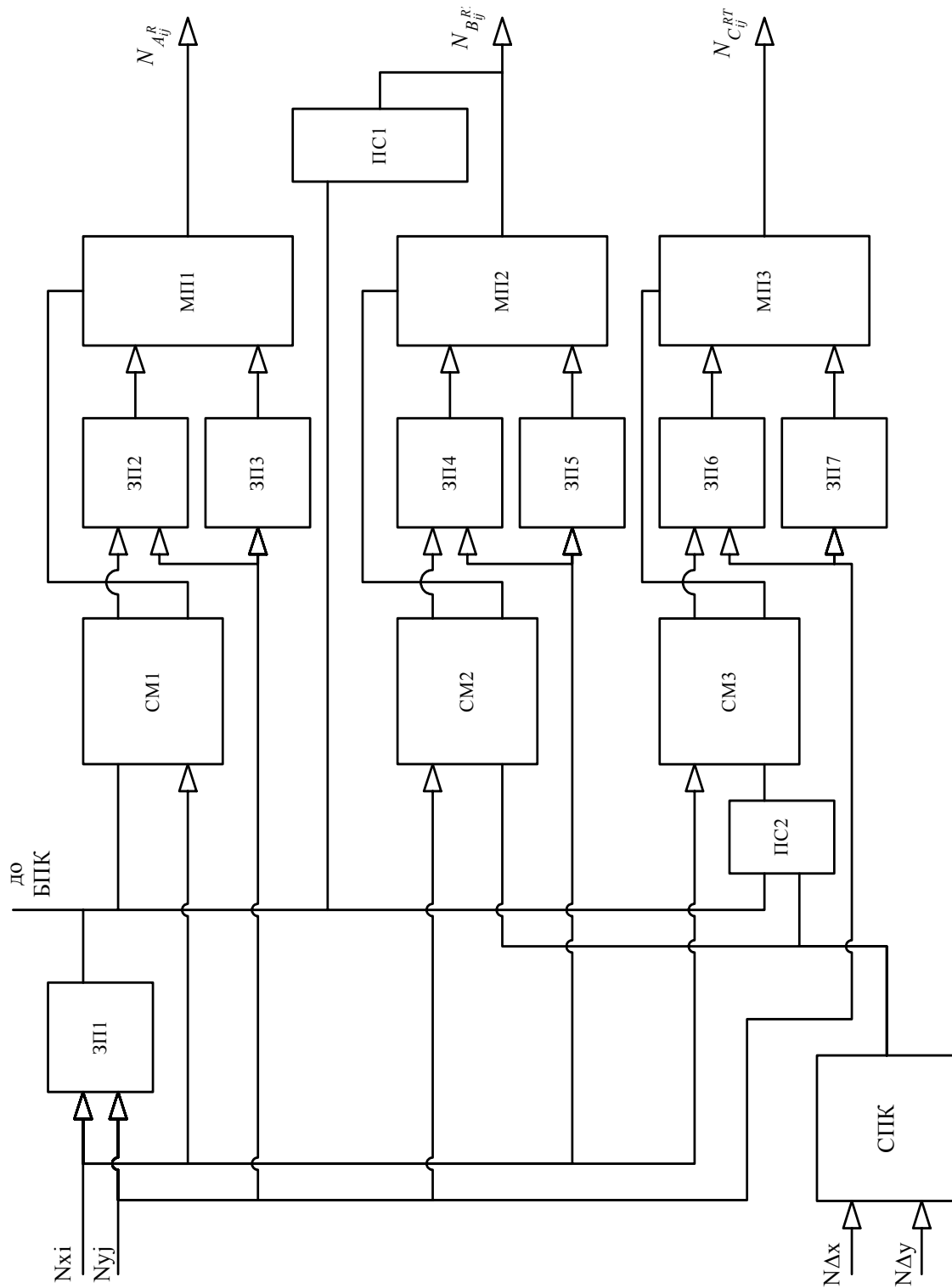


Рисунок 2.26 – Структурна схема блока адресації і блока пам'яті

2.4 Швидка попередня обробка та класифікація зображень дрібноструктурних об'єктів

2.4.1 Вступні зауваження

Надійність і ефективність розв'язання задач розпізнавання або класифікації багатоканальних (зокрема, кольорових оптичних) зображень залежить від ряду факторів. Основними з них є такі:

1) якість вихідних зображень (наявність перешкод, їхній тип і статистичні характеристики; наявність інших видів спотворень, наприклад, розмазування або чіткості границь, точності просторової реєстрації або суміщення компонентних зображень);

2) наявність апріорної інформації про характеристики перешкод або точність їхнього оцінювання за реальним даними;

3) ефективність попередньої обробки зображень (якщо така використовується), погодженість застосовуваних методів попередньої обробки з кінцевими завданнями класифікації багатоканальних зображень;

4) інформативністю ознак, використовуваних для рішення завдання класифікації, відстанню між класами в просторі ознак;

5) повторюваністю статистичних характеристик ознак для різних зображень;

6) наявністю апріорних відомостей про найбільш інформативні ознаки і їхні статистичні характеристики (у найбільш загальному випадку, закон розподілу), а при відсутності таких відомостей – можливістю їхнього одержання за реальним даними й з урахуванням вербального опису класів і їхніх властивостей;

7) ефективністю застосовуваних методів розпізнавання, що залежить від використовуваного математичного апарата й технічних засобів; нерідко застосовність підходу може обмежуватися вимогами до швидкодії виконання класифікації й розмірами зображень.

Для розв'язуваного завдання визначення процентного вмісту матеріалів (Аліт, Беліт, Целіт, Пори) за спеціальним образом підготовленим зображенням зразка характерні такі риси:

1) вихідне зображення (кольорове оптичне) подано в стандартному форматі RGB (три колірних канали), але попередньо стиснутого стандартним JPEG приблизно в 6...6,5 разів; відзначимо, що використання JPEG стиснення під час реєстрації й запису зображень із погляду якості їхньої наступної класифікації є недоцільним, оскільки стиснення із втратами вносить певні спотворення (зокрема, для зображень, стиснутих з пікселями JPEG, після декompresії мають місце блокові ефекти й спотворення середнього рівня в блоках, що може негативно позначитися при застосуванні методів класифікації, заснованих на виявленні меж пікселів об'єктів);

2) візуально помітні певні спотворення, а саме: неоднорідність об'єктів, які необхідно класифікувати; розмитість (нечіткість) границь об'єктів (рис. 2.41); включення фрагментів, подібних іншим класам, усередину пікселів об'єктів (наприклад, Целіту усередині Пір);

3) візуальний аналіз декількох зображень свідчить про приблизну повторюваність колірних ознак об'єктів;

4) апріорна інформація про характеристики перешкод і спотворень відсутня;

5) інформативними ознаками, які можуть використовуватися для розв'язання задачі класифікації, є колірні розходження при поданні вихідних зображень у стандартному форматі RGB; при цьому не ясно, чи є подання вихідних зображень у форматі RGB оптимальним для розв'язання пікселівих задач;

6) інформація про повторюваність статистичних характеристик ознак для різних зображень відсутня; апріорні відомості про статистичні пікселі (закон розподілу) інформативних ознак також відсутні;

7) вербальний опис класів є (класи відповідають чотирьом кольорам, об'єкти площані і їхні розміри, обумовлені кількістю пікселів, не менше пікселів 10x10 пікселів).

Особливістю розв'язуваної задачі класифікації є те, що нетреба пікселювати зображення (визначати замкнуті межі об'єктів і т.д.), а необхідно визначити процентний вміст пікселів, що належать чотирьом класам (фазам) у піксельовому зображенні. Відповідно до зазначеної особливості задачі сегментації не є необхідною, а лише бажаною. Іншими словами, якщо задача визначення процентного вмісту пікселів, що належать чотирьом класам в аналізованому зображенні, може бути вирішена із прийнятною точністю без виконання сегментації, то таке розв'язання може влаштувати. Важливою вимогою є також те, що кожний піксел зображення має бути віднесений до одного з класів.

З урахуванням наведених вище особливостей розв'язуваної задачі, наявної інформації про властивості класів об'єктів і сформованих зображень, доцільним є проведення первинного аналізу властивостей зображень, перешкод і спотворень.

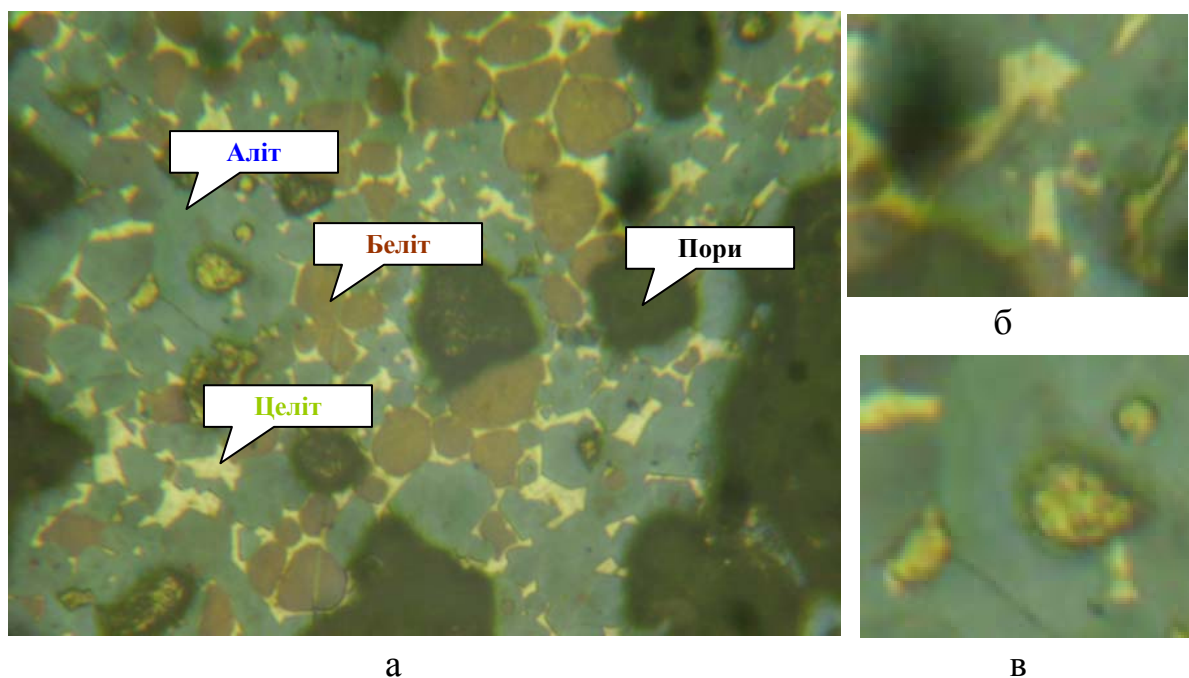


Рисунок 2.41 – Приклад зображення із зазначенням класів фаз (а), розмиті межі об'єктів (б), артефакти (об'єкти жовтих кольорів усередині чорних об'єктів) (в)

2.4.2 Аналіз властивостей зображень, завод і спотворень

Первинний аналіз проводився з використанням як стандартних програмно-алгоритмічних засобів, так і оригінального програмного забезпечення. На першому етапі були в інтерактивному режимі й автоматично визначені характеристики завод для R, G і B компонент декількох кольорових зображень. Основні результати полягають у такому:

– перешкоди мають адитивний характер, закон їхнього розподілу близький до гавсівського (див. один із прикладів гістограми розподілу на рис. 2.42);

– дисперсія цих перешкод σ^2 приблизно постійна й має порядок $6 \dots 8$ (параметр Variance на рис. 2.42) для всіх кольорних компонентів і для всіх досліджених зображень.

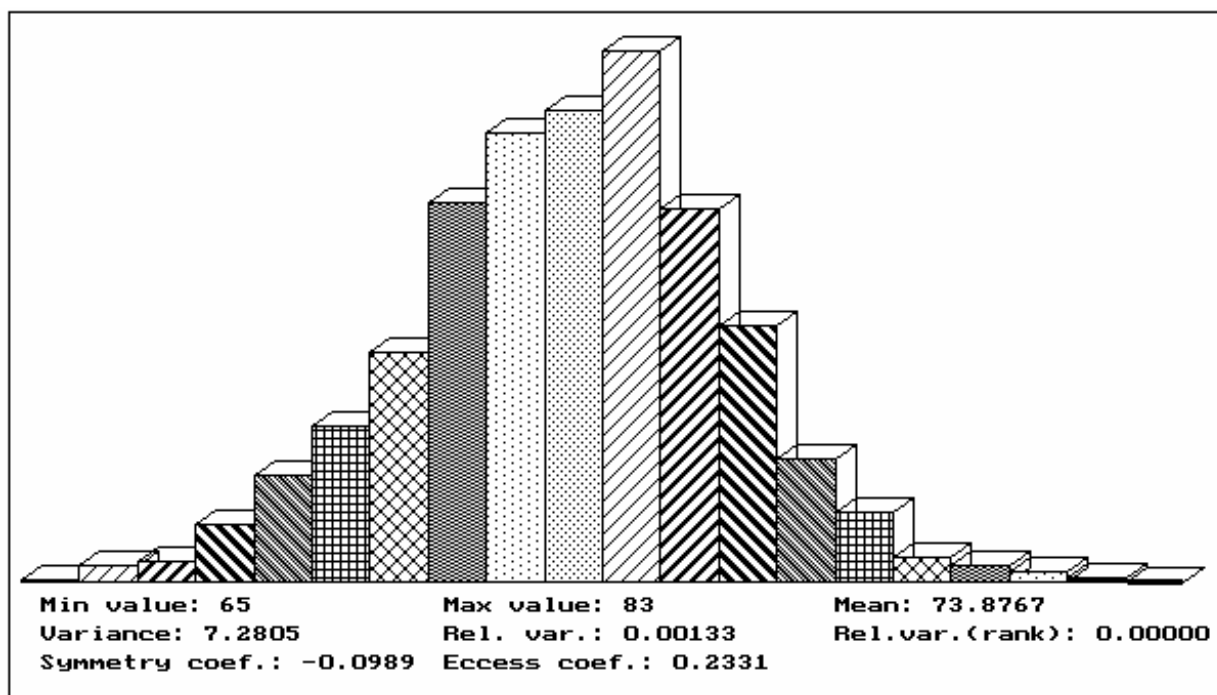


Рисунок 2.42 – Розподіл значень точок у кольорному компоненті R на однорідній ділянці зображення

З погляду класифікації об'єктів у кольоровому зображенні такі характеристики завод є досить сприятливими (набагато гірший варіант - мінливість дисперсії для різних кольорних компонентів і для різних зображень).

На рис. 2.43 для десяти тестових зображень наведені двовимірні гісто-

грами розподілу значень точок, що дозволяють наочніше, ніж на звичайних одномірних гістограмах оцінити наявність і розділювальність класів об'єктів на цих зображеннях.

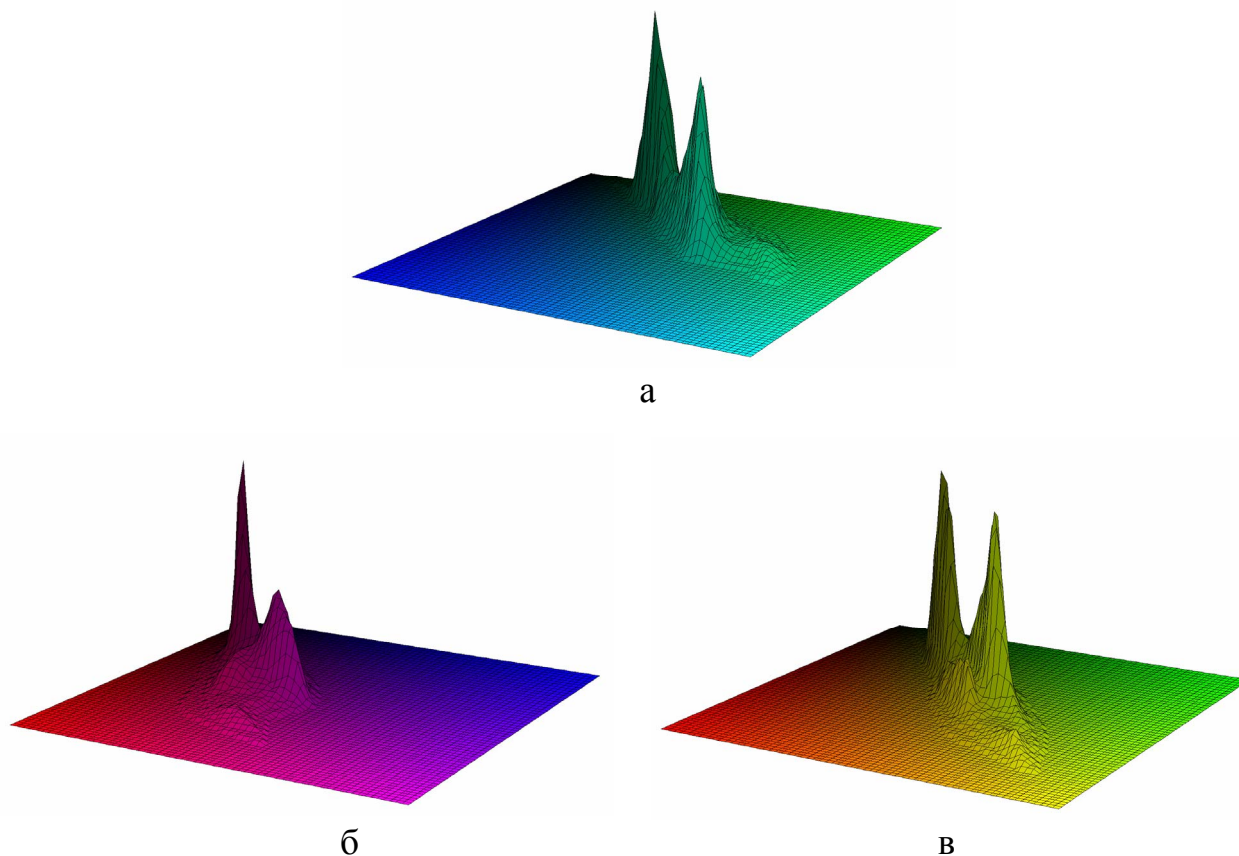


Рисунок 2.43 – Гістограми розподілу значень точок на зображеннях, що класифікуються, у просторах: а) G-B, б) R-B, в) R-G.

Добре видно наявність декількох піків на гістограмах, що свідчать про наявність широко виражених класів об'єктів на досліджуваних зображеннях. У той же час чітких меж між цими класами не спостерігається, що говорить про складності задачі класифікації ділянок зображень.

Важливі також просторові спектральні або кореляційні характеристики завад. Їхнє дослідження показало, що існує досить істотна кореляція флуктуацій значень сусідніх точок (див. перетини двовимірної АКФ на рис. 2.44). Це свідчить про те, що завади не можна вважати некорельованими (білим шумом).

Серед характерних спотворень, виявлених у результаті аналізу зображень, відзначимо два основних:

1) «розмитість» меж площаних об'єктів, яка може бути обумовлена або самими властивостями зображень, або тим, що використовуване при формуванні зображень просторове розділення «непогоджено» з розмірами пікселів;

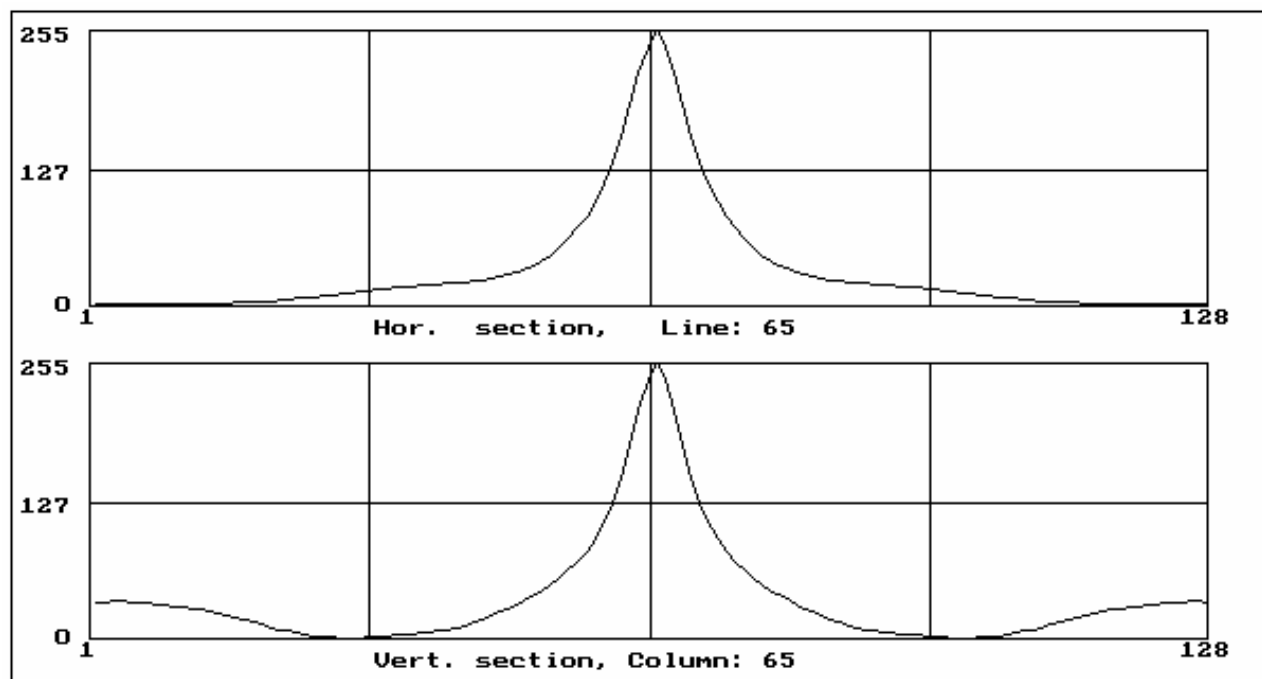


Рисунок 2.44 – Перетини автокореляційної функції по горизонталі й вертикалі

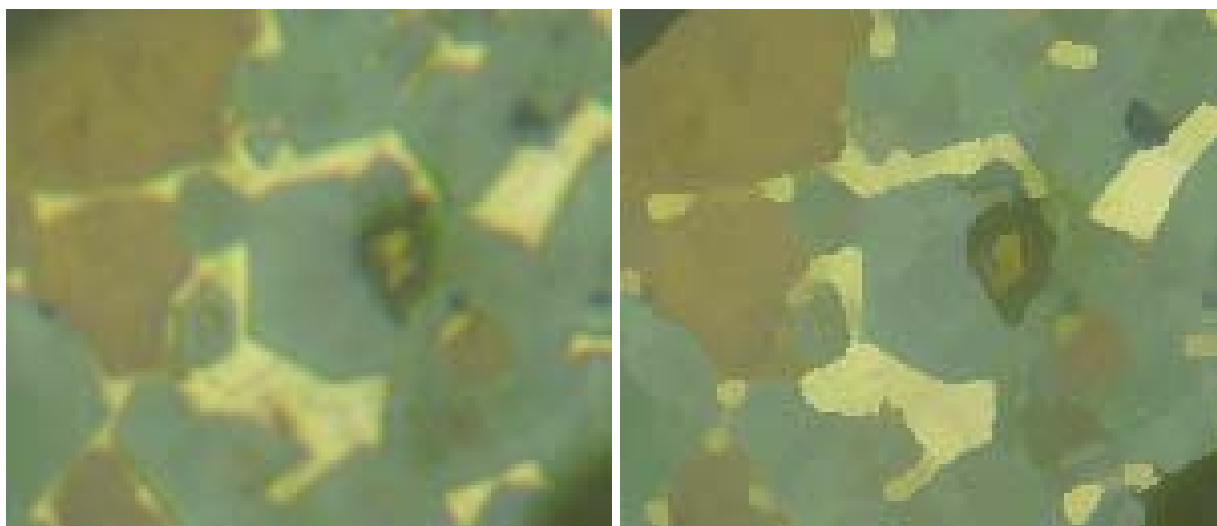
2) невеликий зсув зображення зеленого колірному компоненту щодо червоного і синього (ефект розшарування кольорів, що досить часто зустрічається в практиці формування RGB зображень, див. рис. 2.45, а).

Ширини перехідних областей між площаними об'єктами становлять 2...4 пікселів, що є корисною інформацією при виборі параметрів методів підкреслення меж і усунення ефектів розшарування кольорів при обробці кольорових зображень.

Як показує досвід розв'язання аналогічних задач, застосування методів попередньої обробки кольорових зображень із метою підкреслення меж звичайно приводить до підвищення надійності класифікації або виділення певного типу об'єктів за кольорними ознаками.

Тому була розглянута можливість використання такої обробки із застосуванням модифікації методу.

Очевидно, що вдалося досягти більшої однорідності площаних об'єктів (часткового заглушення завад) і більш різких меж площаних об'єктів.



а

б

Рисунок 2.45 – Розшарування кольорів на краях об'єктів і зсуву зеленого колірний компонента щодо червоного й синього (зеленувата облямівка уздовж нижніх меж об'єктів типу «Беліт» і червонувата – уздовж верхніх меж) (а), результат векторного підкреслення меж для цієї ж ділянки (б)

Виходячи із проведеного статистичного аналізу вхідних даних (зображень), можна рекомендувати таке.

Не використовувати формат JPEG для стиснення зображень. Використання стиснення із втратами інформації, яким є JPEG, у цьому випадку невіправдано, тому що немає необхідності в економії місця на жорсткому диску, що займане аналізованими зображеннями (невеликими за розміром). Для кодування зображень необхідно використовувати один з форматів TIFF або RAW, якщо вони доступні для використовуваної моделі цифрового фотоапарата (для багатьох фотоапаратів фірми Olympus формат RAW доступний). При неможливості використовувати інший формат, крім JPEG, виставляти мінімальний стиснення у налаштуваннях фотоапарата.

Використання більш високого розділення фотоапарата, ніж 640x480, у цьому випадку недоцільно. Аналіз функції автокореляції показує високу розмитість точок зображення (приблизно 2-3 точки зображення відповідають одному елементу розділення). Як ми вже відзначали, це може бути пов'язане з особливостями апаратури (мікроскопа, оптики), так і із властивостями зображень. Збільшення розділення знімків може привести тільки до зростання часових витрат на їхню класифікацію без збільшення точності такої класифікації.

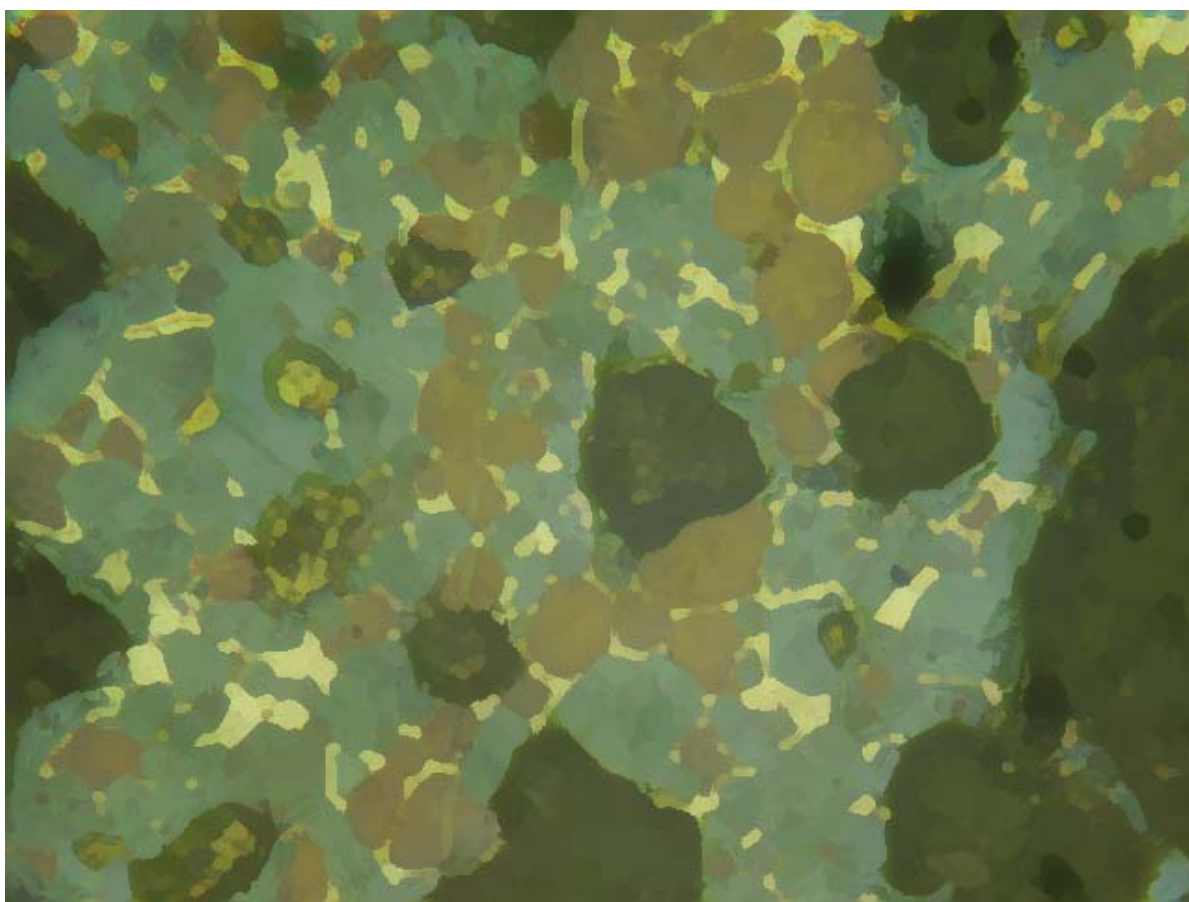


Рисунок 2.46 – Ілюстрація застосування векторного фільтра для усунення ефектів розшарування кольорів на краях об'єктів і взаємних зсувів колірних компонентів (стосовно до зображення на рис. 2.41)

У зв'язку з істотною флуктуацією яскравості зображень, а також зміною яскравості об'єктів одного класу усередині кожного зображення поряд зі значеннями яскравості колірних компонентів R, G і B при класифікації рекомендується використовувати й співвідношення кольорів. Наприклад, для цьо-

го можна використовувати колірний простір YUV, у якому компонента Y відповідає яскравості зображення, а компоненти U і V – співвідношенням кольорів. Вирази для перетворення зображення з RGB в YUV і зворотно наведені нижче.

$$R = Y + 1,4075 \cdot (V - 128);$$

$$G = Y - 0,3455 \cdot (U - 128) - 0,7169 \cdot (V - 128);$$

$$B = Y + 1,7790 \cdot (U - 128);$$

$$Y = R \cdot 0,299 + G \cdot 0,587 + B \cdot 0,114;$$

$$U = -R \cdot 0,169 - G \cdot 0,332 + B \cdot 0,500 + 128;$$

$$V = R \cdot 0,500 - G \cdot 0,419 - B \cdot 0,0813 + 128;$$

На рис. 2.47 наведено зображення у форматі YUV і окремо по колірних шарах Y, U і V (помітне розшарування кольорів у вигляді тіней об'єктів).

Для усунення адитивного шуму на зображеннях рекомендується попередня фільтрація, яку необхідно здійснювати до класифікації. При цьому для більшості ефективних фільтрів як вхідний параметр необхідно вказати дисперсію шуму. У цьому випадку можна завжди вказувати фіксоване, оцінене по досліджуваних зображеннях значення 7, а можна й використовувати методи автоматичної оцінки дисперсії шуму для кожного зображення, що класифікується, окремо.

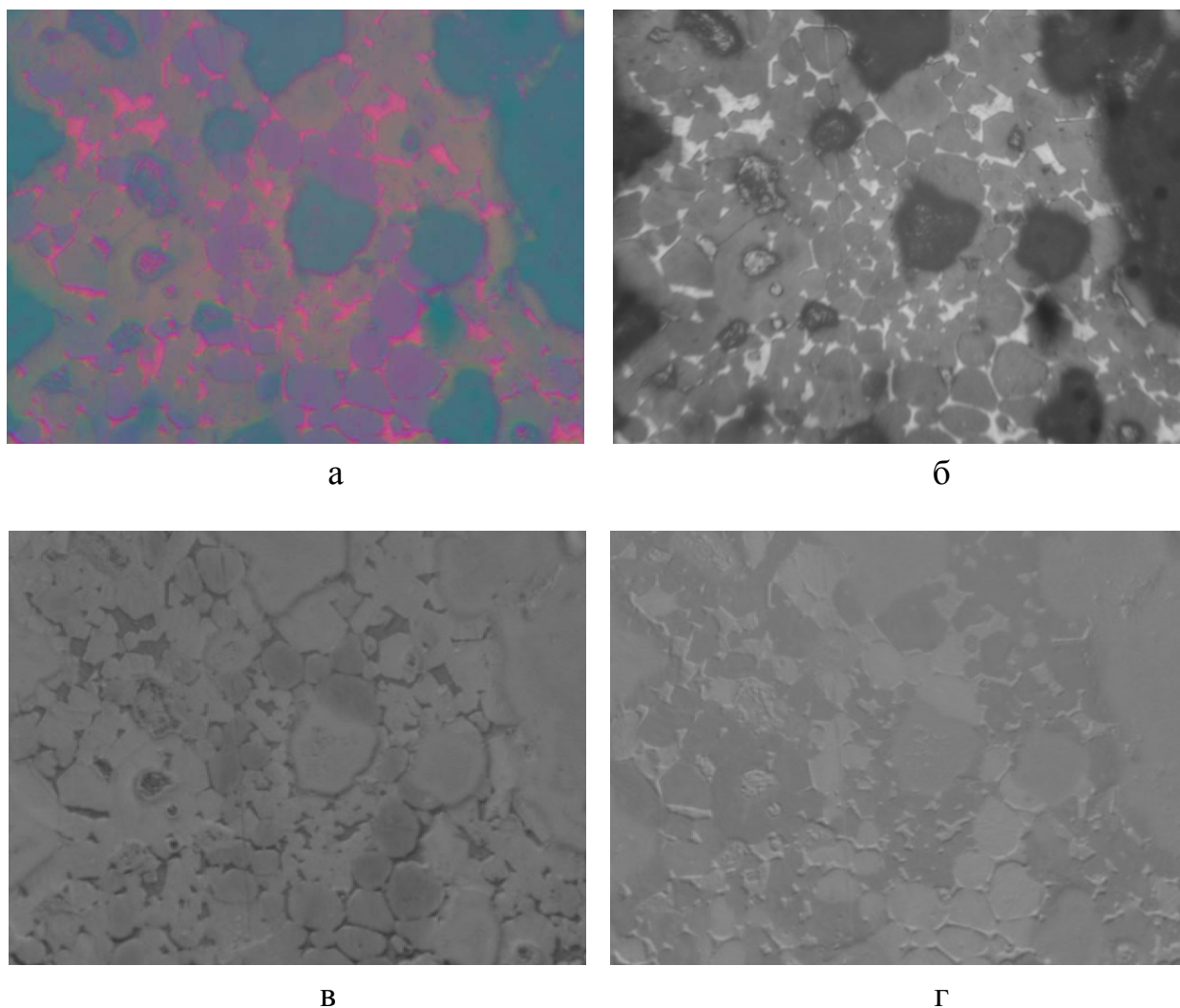


Рисунок 2.47 – Зображення з рис. 2.41, подане в колірному просторі YUV (а), колірний шар Y (б), колірний шар U (в), колірний шар V (г)

Для усунення зсуву колірних шарів і розшарування кольорів на межах об'єктів рекомендується використовувати векторне підкреслення меж зображень. Причому, на відміну від фільтрації для усунення адитивного шуму на зображеннях, результат векторного підкреслення меж рекомендується використовувати не замість, а поряд з вихідним зображенням при класифікації об'єктів зображення.

У зв'язку з істотним перетинанням класів «Аліт», «Беліт», «Целіт» і «Пори», і теоретичною неможливістю абсолютно точного розподілу цих класів, рекомендується при класифікації об'єктів використовувати не жорстку, а гнучку логіку, що може трохи підвищити точність обчислення сумарних площ об'єктів.

2.4.3 Аналіз ознак об'єктів і вимоги до методу класифікації

Логічно припустити, що основними ознаками, які повинні використовуватися при класифікації об'єктів, є колірні. Такими можуть бути як власне інтенсивності (значення) кожної з колірних компонентів, так і їхні стосунки (компоненти U і V у просторі YUV), які більш інваріантні до умов освітлення, при яких провадилося формування зображення, чим власне інтенсивності.

Аналіз признакового простору провадився за допомогою спеціально розробленого програмного забезпечення, що дозволяє візуалізувати розподіли значень для двох будь-яких пар колірних компонентів. Отримані гістограми наведені на рис. 2.48 (пояснення дані в підписові).

Як витікає з аналізу гістограм, наведених на рис. 8, для класів “пори” і “целіт” двовимірні розподіли мають практично одномодальний характер. У той же час, для інших класів “аліт” і “беліт” характерні багатомодальні розподіли, що для традиційних класифікаторів, наприклад, Байєса, призводить до значного зниження їхньої ефективності. Крім того, простори ознак для всіх класів є негавсівськими, що додатково обмежує застосовність традиційних класифікаторів і знижує їхню ефективність. Відстані між центрами класів також різні.

Нарешті, розподіли значень явно перетинаються. Це свідчить про те, що ніякі алгоритми класифікації, засновані на використанні правил типу “якщо $L < R < U$, то віднести даний піксел до k -го класу” (де L і U – відповідно нижній і верхній поріг, R – значення компонента) або їхніх комбінацій, не здатні добре розділити класи. Більш того, застосування класифікації на основі таких порівняно простих правил неминуче призведе до того, що досить значне число пікселів не будуть віднесені до того або іншого класу.

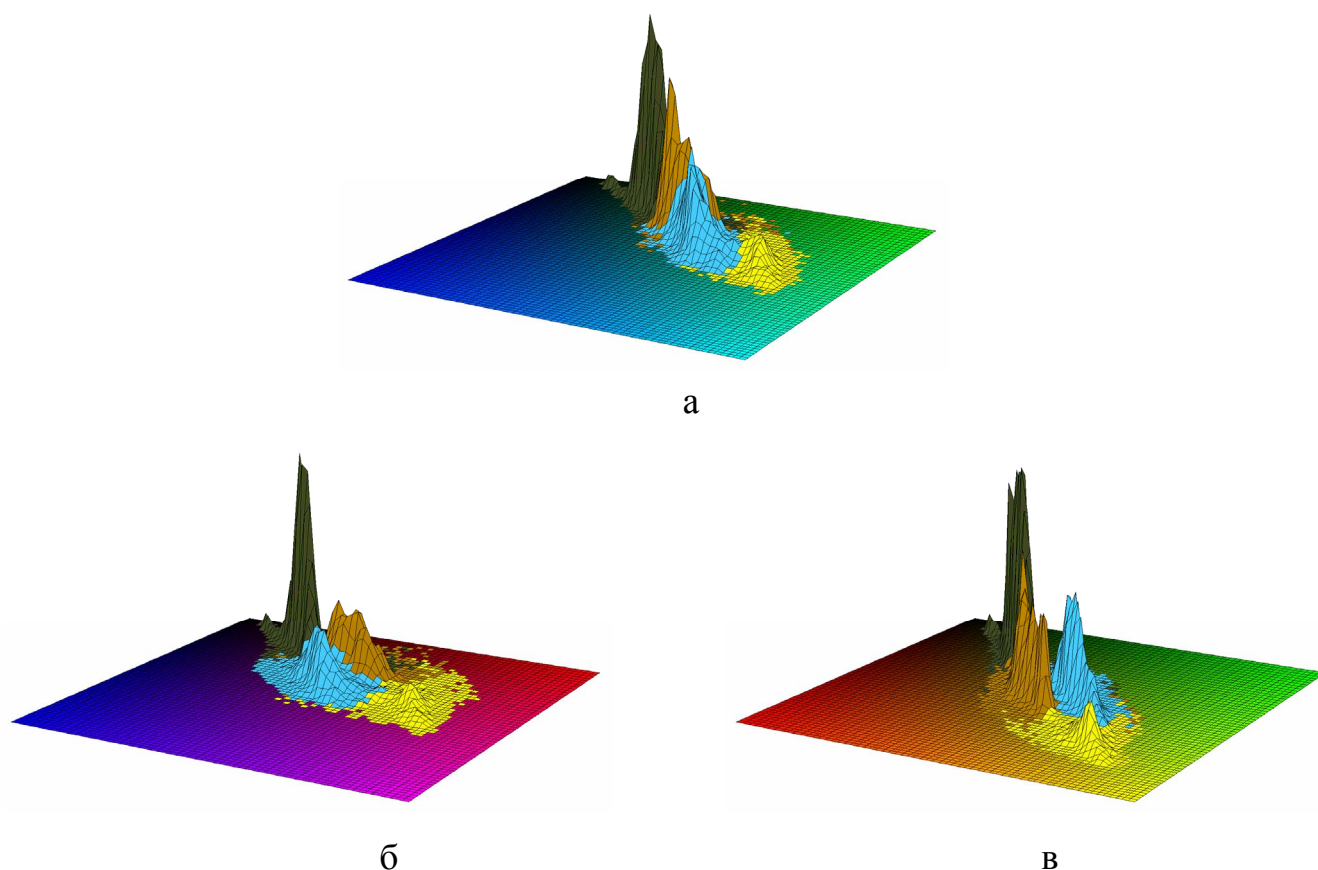


Рисунок 2.48 – Гістограми розподілу значень точок по типах об'єктів у просторах: а) G-B, б) R-B, в) R-G. Типи об'єктів показані: аліт – блакитним кольором, беліт – коричневим кольором, целіт – жовтим кольором, пори – темним кольором

Центри класів (середні значення) по компонентах R, G і B наведені в таблиці 2.6. Дані отримані в результаті аналізу тестових вибірок для чотирьох зображень.

Ґрунтуючись на проведеному аналізі, до методу класифікації можна пред'явити такі основні вимоги:

Метод класифікації має бути здатний ефективно враховувати не 3, а досить велику (10 і більше) кількість інформативних ознак.

Метод класифікації поряд з колірними ознаками має уміти враховувати просторову інформацію.

Таблиця 2.6 – Центри класів по компонентах R, G і B

Аліт			Беліт			Целіт			Пори		
R	G	B	R	G	B	R	G	B	R	G	B
115	141	94	117	114	68	174	182	103	69	78	43

У зв'язку з неочевидністю вибору набору інформативних ознак для класифікації, метод класифікації має бути стійкий до наявності фіктивних і надлишкових ознак. Під фіктивними розуміються неінформативні або шумоподібних ознаки. Під надлишковими розуміються ознаки, в істотному ступені продубльовані іншими ознаками з використовуваного набору. У процесі настроювання класифікатора має використовуватися зазначений оператор набір еталонних ділянок (навчальна вибірка).

- 1) Для усунення артефактів (рис. 2.41, в) необхідно передбачити морфологічну обробку результатів класифікації (може використовуватися опціонально). Метод класифікації поряд з колірними ознаками має уміти враховувати просторову інформацію.
- 2) У зв'язку з неочевидністю вибору набору інформативних ознак для класифікації, метод класифікації має бути стійкий до наявності фіктивних і надлишкових ознак. Під фіктивними розуміються неінформативні або шумоподібних ознаки. Під надлишковими розуміються ознаки, в істотному ступені продубльовані іншими ознаками з використовуваного набору. У процесі настроювання класифікатора має використовуватися зазначений оператор набір еталонних ділянок (навчальна вибірка).

3) Для усунення артефактів (рис. 2.41, в) необхідно передбачити морфологічну обробку результатів класифікації (може використовуватися опціонально).

4) Для усунення артефактів (рис. 2.41, в) необхідно передбачити морфологічну обробку результатів класифікації (може використовуватися опціонально).

2.4.4 Пропонований підхід до розв'язання задачі класифікації

Для розв'язання даної задачі класифікації найбільш доцільним і виправданим нам представляється підхід, структурна схема якого представлена на рис. 2.49.

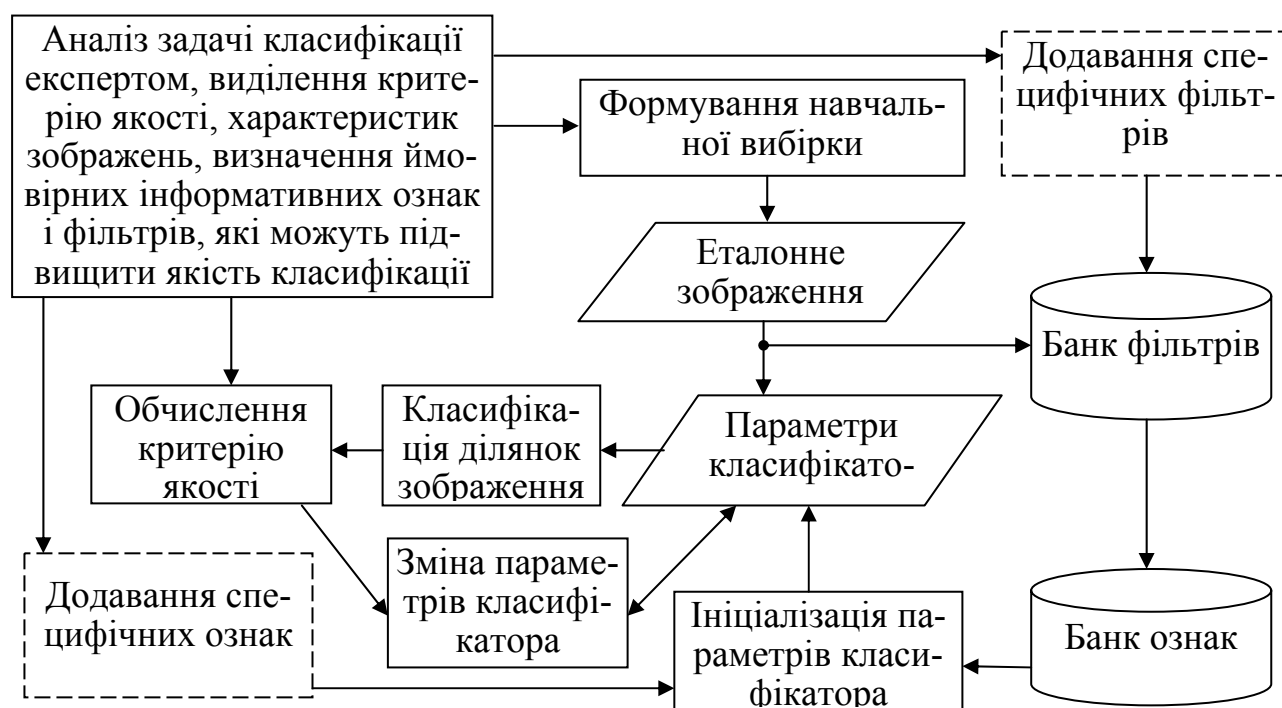


Рисунок 2.49 – Структурна схема калібрування (навчання) класифікатора

Навчання при даному підході є ітеративним з використанням орієнтованої на мінімізацію (максимізацію) заданого критерію якості кластеризації. Цей підхід може використовуватися як для ефективної класифікації ділянок зображень, так і для ефективного вибору методів подальшої обробки (аналізу) цих ділянок, наприклад, фільтрації з метою заглушення шумів.

Пропонований підхід стійкий до наявності в банку фіктивних і надлишкових ознак. Збільшення банку ознак призводить у найгіршому разі до

прямо пропорційного зростання часу навчання, не погіршуючи якості навчання, а також, не впливаючи на час, затрачуваний на класифікацію за допомогою відкаліброваного (навченого) класифікатора.

У прототипі системи розпізнавання, реалізованого у формі програмного забезпечення для формування навчальної вибірки був використаний наступний простий механізм. Оператор вибирає довільну ділянку прямокутної форми на зображенні (рис. 2.50), указуючи для нього один із чотирьох типів (Аліт, Беліт, Целіт або Пори).

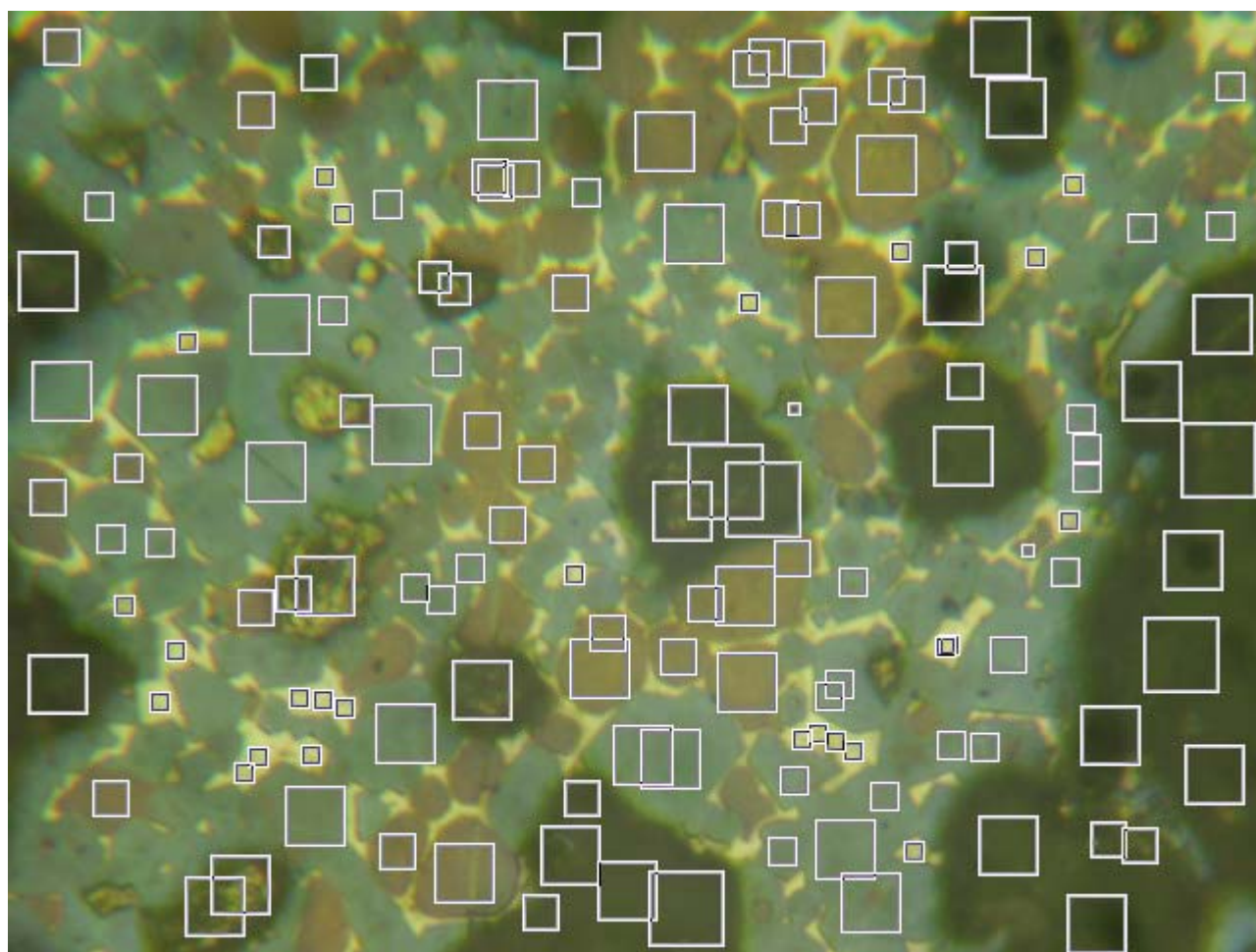


Рисунок 2.50 – Вибір еталонних фрагментів зображення для калібрування класифікатора

При цьому вся прямокутна ділянка має припадати на об'єкт одного типу. Відповідно для, наприклад, пор, що займають відносно великі по площі ділянки, зручно використовувати прямокутники великого розміру. У той же

час для зазначення зразків ділянок Целіту доводиться використовувати прямокутники меншого розміру.

Навчальна вибірка обов'язково повинна містити ділянки всіх типів об'єктів, що класифікуються. Ділянки можуть бути зазначені з певною погрішністю (рамка ділянки може заходити на об'єкт іншого типу). Чим менше така погрішність, тим вище точність калібрування класифікатора.

Описаний механізм при своїй простоті, має й недоліки. Для оператора дуже непросто вказувати прямокутником краї об'єктів (найбільш складні для класифікації й важливі для навчання) щоб не захопити при цьому об'єкти іншого типу. При розробці промислового варіанта програмного забезпечення необхідно вдосконалити механізм вибору навчальних ділянок, щоб спростити роботу оператора й підвищити точність калібрування класифікатора.

Гістограми розподілу значень яскравостей точок по отриманій вибірці наведені на рис. 2.48.

У банк ознак було включено 19 колірних і просторових ознак, з яких класифікатором у процесі калібрування 3 ознаки були відсіянні як неінформативні. Не зупиняючись на описі включених у класифікацію ознак, наведемо їхні відносні значимості, визначені в процесі калібрування класифікатора (див. таблицю 2.7). Видно наявність принаймні 8 досить інформативних ознак, значимість яких перебуває на рівні 5% і вище.

Таблиця 2.7 – Відносні значимості згідно з калібруванням

Ознака	1	2	3	4	5	6	7	8
Значимість, %	4,9	2,4	2,4	0,8	15,8	2,7	2,5	1,0
Ознака	9	10	11	12	13	14	15	16
Значимість, %	13,2	0,6	13,5	7,4	15,5	7,3	1,6	8,4

Усього ж нам представляється доцільним включення в банк ознак ще приблизно 40-50 інтегральних і просторових ознак.

У результаті калібрування класифікатора відповідно до пропонованого підходу, крім ряду другорядних показників, обчислюються два дуже

важливих показники якості калібрування, що дають можливість операторові отримати представлення про якість вибору їм навчальної вибірки й/або про принципову можливість розподілу використаних зображень на класи за допомогою даного набору ознак. Цими показниками якості є середня впевненість у правильній класифікації точки й прогнозоване середнє відхилення отриманих у результаті класифікації значень площ об'єктів від реальних значень.

Для розглянутого прикладу середня впевненість у правильній класифікації точки склала 95,612%, а прогнозоване середнє відхилення отриманих у результаті класифікації значень площ об'єктів від реальних значень – 0,537%. Для підвищення першого показника й зменшення другого в рамках запропонованого підходу існують два шляхи: збільшення банку ознак і підвищення якості вибору оператором навчальної вибірки.

2.5. Структури засобів обробки великих масивів інформації

2.5.1. Апаратна реалізація нейронної мережі СМАС: загальний підхід

Апроксимація функцій будь-якого вигляду нейронною мережею СМАС зводиться до виконання таких операцій:

- кодування змінних, в результаті якого формуються вектор асоціацій, що є адресою читання ваг з оперативної пам'яті, і вектор коефіцієнтів, – значень активаційних функцій вхідного шару нейронів, що беруть участь у формуванні чергового значення функції;
- хешування вектора асоціацій, яке служить для скорочення об'єму пам'яті, необхідного для зберігання ваг;
- вибірка значень ваг і обчислення значення функції;
- порівняння набутого значення функції з потрібним і коректування ваг за одним з алгоритмів навчання.

Кодування змінних. У цифровій системі обробки інформації змінні подаються у формі двійкових слів. Вважатимемо, що змінні x_i , від яких залежить функція y , мають однакову розрядність, рівну n . Вхідний шар мережі містить ρ ступенів квантування, таких, що для будь-якого ступеня існує інший ступінь, на якому області квантування зміщені щодо даного ступеня на один рівень квантування вхідної змінної x_i . Оскільки ця змінна є двійковим числом, таке зміщення відповідає одиниці молодшого розряду вхідної змінної. Привласнимо ступені квантування змінної x_i індекс $j_i = \overline{0, \rho - 1}$.

Областю квантування є нейрон, що має унікальний для даного ступеня номер $NNS_i^{j_i}$ (Neuron Number on Stair j_i , variable i) і функцію активації $\Phi(u_i^{j_i})$. Область визначення цієї функції $u_i^{j_i} \in [0, \rho - 1]$, оскільки область квантування містить ρ рівнів квантування вхідної змінної. Приймемо, що всі нейрони вхідного шару мають функції активації одного вигляду. Аргументом цієї функції буде значення

$$u_i^{j_i} = (x_i + j_i) \bmod \rho, \quad (2.100)$$

де mod – функція обчислення залишку від цілочислового ділення.

Номер активного нейрона на ступені j_i можна визначити як:

$$NNS_i^{j_i} = (x_i + j_i) \text{div } \rho, \quad (2.101)$$

де div – функція цілочислового ділення.

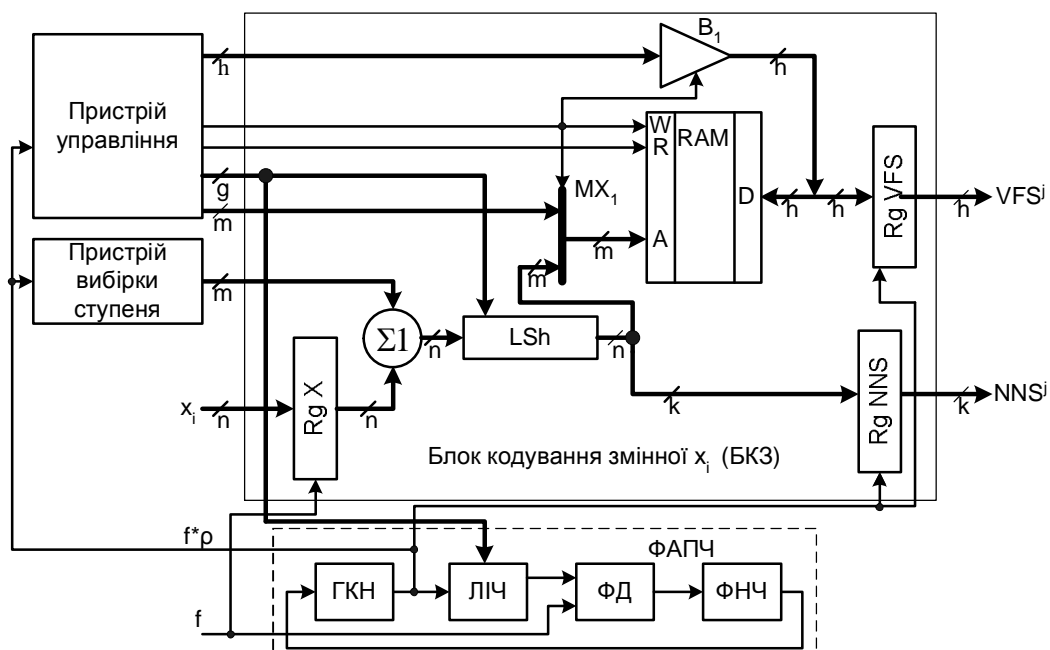
Апаратна реалізація операції ділення на довільне число є або ресурсомісткою, або повільною. При виборі значень ρ , рівних 2^m , операцію ділення можна замінити операцією зсуву на m розрядів управо. Залишком від ділення, в цьому випадку, є m молодших розрядів вхідної змінної, які можуть використовуватися як аргумент активаційної функції.

У класичному варіанті мережі СМАС активаційні функції нейронів областей квантування мають прямокутну форму. Проте, як показують дослідження, кращий результат досягається під час використання функцій вищих порядків (лінійних, квадратичних, кубічних), а також тригонометричних функцій і функцій степені. Апаратна реалізація широкого спектра активаційних функцій (яка необхідна для структури, що моделює мережу СМАС) складна і ресурсоемна. Тому пропонується таблично задавати функцію активації в ОЗП, що містить 2^m комірок. Розрядність комірок ОЗП h визначається областю допустимих значень функції активації, а також необхідною точністю задання значень функції активації. Наприклад, якщо $m = 8$, то необхідне ОЗП місткістю 256 комірок. Блоки ОЗП такого об'єму є в більшості сучасних ПЛІС, наприклад, сімейства АСЕХ1К фірми ALTERA. Така реалізація функцій активації є простою і гнучкою, оскільки дозволяє задати функцію будь-якого вигляду.

Структуру блока кодування змінної x_i наведено на рис. 5.31. На вхід цього блока значення змінної поступають в паралельному коді з частотою f . Для кожного значення, що поступило, необхідно отримати ρ значень NNS і функцій активації. Тобто кодування має здійснюватися з частотою $f \cdot \rho$. Таку частоту мож-

на отримати за допомогою схеми фазового автоматичного підстроювання частоти (ФАПЧ), що складається з генератора, керованого напругою (ГКН), фазового детектора (ФД), фільтра низьких частот (ФНЧ). Напруга, ГКН, що управляє, залежить від того, скільки періодів формованої частоти потрапило в один період опорної. Воно формується ФД, який порівнює фази опорної частоти f і сигналу переповнювання лічильника з модулем ρ , що підраховує імпульси частоти, яка синтезується. Напруга управління ГКН фільтрується за допомогою ФНЧ.

Код змінної x_i зашипується в регістрі RGX, звідки поступає на суматор $\Sigma 1$. На другий вхід цього суматора поступає m -розрядне значення номера ступеня j , на якому в даний момент відбувається кодування. Це значення формується з частотою $f \cdot \rho$ пристроєм вибірки ступеня, який служить для усунення діагональної структури асоціативного поля, і, відповідно, поліпшення апроксимуючих властивостей мережі. З виходу суматора n -розрядний код поступає на логічний зрушувач (LSh – Logical Shifter), що дозволяє змінювати значення m . При фіксованому значенні m логічний зрушувач як такий зникає. Значення u_i^j визначають m молодших розрядів вихідного коду LSh, а значення $NNS_i^{j_i}$ – відповідно старші $n - m$ розрядів. Значення $u_i^{j_i}$ поступає на адресні входи ОЗП зберігання активаційної функції. З виходу ОЗП h -розрядне значення функції активації поступає на регістр RgVFS, а з регістра – на вихід VFS (Value of Function on Stair) блока кодування. Значення $NNS_i^{j_i}$ поступає на регістр RgNNS, а звідти на вихід NNS (Number of Neuron on Stair) блока кодування. Тактові імпульси на регістри, а також сигнали управління ОЗП формуються з частотою $f \cdot \rho$ пристроєм управління, загальним для всієї структури. Мультиплексор MX_1 і буфер B_1 необхідні для ініціалізації ОЗП значеннями активаційної функції. У разі фіксованого виду цієї функції необхідність в цих блоках відпадає.

Рисунок 5.31 – Блок кодування змінної x_i

При малих значеннях m (4–6) можлива реалізація активаційної функції безпосередньо на логічних елементах ПЛІС, що є так званими пошуковими таблицями LUT (Look-up Table). Зокрема, для вже відміченого сімейства ПЛІС ACEX1K LUT дозволяє реалізувати будь-яку двійкову функцію від 4 двійкових змінних. Так, при $m=4$ і 8-розрядному поданні активаційної функції буде потрібно всього 8 комірок ПЛІС. Функція синтезується на етапі проектування спецпроцесора і її не можна змінити під час його роботи.

Блоки кодування змінних працюють паралельно і формують вектор асоціацій і вектор коефіцієнтів, компоненти яких послідовно розподілені в часі. Вектор асоціацій можна подати у вигляді:

$$a_k = \begin{pmatrix} 0 : NNS_r^0 : \dots : NNS_2^0 : NNS_1^0, \\ 1 : NNS_r^1 : \dots : NNS_2^1 : NNS_1^1, \\ \dots, \\ (\rho - 1) : NNS_r^{\rho-1} : \dots : NNS_2^{\rho-1} : NNS_1^{\rho-1} \end{pmatrix}, \quad (2.102)$$

де k – номер вхідної комбінації змінних; r – кількість вхідних змінних.

Залежність (2.102) відображує тільки ненульові компоненти вектора асоціацій і визначає, до яких елементів пам'яті ваг буде проведено звернення під час обчислення функції.

Елемент вектора асоціацій є комбінацією номера кроку кодування і номерів нейронів на ступенях, що беруть участь в кодуванні на даному кроці. Кількість кроків кодування рівна кількості ступенів ρ . Проте номер кроку не обов'язково відповідає номеру ступеня для i -ї змінної. Як вже було зазначено, для усунення діагональної структури асоціативного поля ступеня із зрушеними на один рівень квантування областями можуть знаходитися на послідовних кроках кодування тільки у однієї із змінних. Для цієї змінної номер кроку кодування рівний номеру ступеня j . Для решти змінних номера ступенів, які беруть участь в кодуванні на даному кроці, формує пристрій вибірки ступеня.

Розрядність q_{av} елемента вектора асоціацій рівна:

$$q_{av} = m + r(n - m), \quad (2.103)$$

де m , n і r визначено вище.

Вважатимемо, що змінні подаються однаковою кількістю розрядів.

Значення q_{av} визначить об'єм пам'яті V_{wc} , необхідний для зберігання вагових коефіцієнтів мережі:

$$V_{wc} = 2^{q_{av}}. \quad (2.104)$$

Елементи вектора коефіцієнтів формуються як добуток значень активційних функцій для всіх змінних на даному кроці кодування. Весь вектор коефіцієнтів (Ψ_k) має вигляд:

$$\Psi_{\mathbf{k}} = \begin{pmatrix} \Phi(u_r^0) * \dots * \Phi(u_2^0) * \Phi(u_1^0), \\ \Phi(u_r^1) * \dots * \Phi(u_2^1) * \Phi(u_1^1), \\ \dots, \\ \Phi(u_r^{\rho-1}) * \dots * \Phi(u_2^{\rho-1}) * \Phi(u_1^{\rho-1}) \end{pmatrix}. \quad (2.105)$$

Розрядність елементів вектора коефіцієнтів на відміну від вектора асоціації визначається не тільки розрядністю значень $\Phi(u_i^{j_i})$, яка рівна h . Це відбувається унаслідок того, що область допустимих значень активаційної функції нормується на інтервалі $[0...1)$. Тобто є дробовим числом. При цьому недоцільно використовувати розряд для цілої частини, і всі h розрядів відводяться під дробову частину значення активаційної функції. Добуток значень, що лежать в інтервалі $[0...1)$, також лежить в інтервалі $[0..1)$. Кількість розрядів q_{cv} , необхідне для подання цього здобутку, визначається бажаною точністю обчислень. Максимальна точність буде при $q_{cv} = r \cdot h$.

Під час використання мережі СМАС для обробки зображень, кількість вхідних змінних визначається або кількістю координат (дві) точки, або кількістю кольорних складових (три) точки в зображенні. Крім того, необхідність виконання ρ кроків при кодуванні для кожної точки збільшує необхідну для роботи тактову частоту. Так, при $\rho=32$ і частоті проходження точок у розгортці зображення $f=13,5$ МГц, значення робочої тактової частоти має складати $f \cdot \rho = 13,5 \cdot 32 = 432$ МГц. Ця величина є граничною для доступних на сьогоднішній день ПЛІС, а, отже, розгляд значень ρ , більш ніж 32, позбавлено прикладного сенсу.

Для формування значень вектора коефіцієнтів необхідна реалізація операції множення, витрати на яку великі і пропорційні кількості множників. Водночас множники є вибраними з ОЗП значеннями базисної функції. Розрядність адрес для звернення до цього ОЗП рівна m . При $\rho = 32$, $m = 5$. Тобто ОЗП містить всього 32 комірки. Пропонується замінити обчислення добутку базисних функцій вибіркою значень цього добутку із загального для всіх змінних ОЗП,

яке міститиме ρ^N комірок. При кількості змінних $N = 2$ і $\rho = 32$ об'єм ОЗП складе 1024 комірки, при $N = 3$ – 32768 комірок. ОЗП такого об'єму є в структурах приступних ПЛІС. Перевагами такої модифікації є:

1. Зниження апаратних витрат, пов'язане з усуненням помножувача і структур для ініціалізації ОЗП базисних функцій для кожної змінної.
2. Усунення похибок, пов'язаних з обмеженою розрядністю окремих ОЗП і помножувача.
3. Скорочення об'єму обчислень при кодуванні змінних.

Обчислення значення функції. Сформований в результаті кодування вектор асоціацій за необхідності обробляється алгоритмом хешування, в результаті роботи якого формується вектор адрес a'_k вибірки ваг з пам'яті. В результаті вибірки з пам'яті отримаємо вектор ваг $w_k(a'_k)$, який спільно з вектором коефіцієнтів Ψ_k визначить значення функції для даної комбінації вхідних:

$$y_k = w_k^T(a'_k)\Psi_k. \quad (2.106)$$

На рис. 2.51 наведено загальну структуру апаратної реалізації мережі СМАС, що містить дві вхідні змінні.

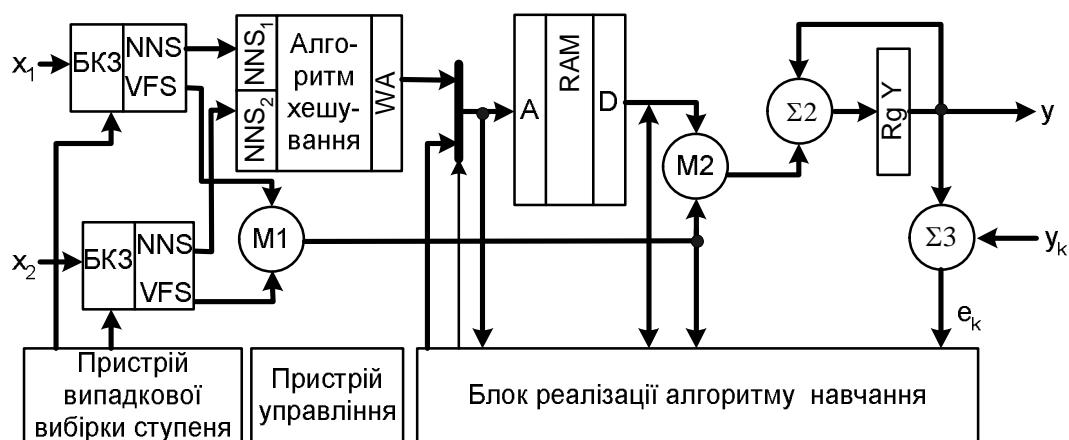


Рисунок 2.51 – Загальна структура апаратної реалізації мережі СМАС

Мережа обробляє дві вхідні змінні, кожна з яких кодується блоками БКП, що містять рівну кількість ступенів квантування. Функціонування такої структури полягає в такому. На j -му такті блоки БКЗ формують значення номерів нейронів у поточних ступенях (NNS_1 і NNS_2) і значення їх функцій активації (VFS_1 і VFS_2). Значення VFS поступають на помножувач $M1$, що формує елементи вектора коефіцієнтів (2.105). Значення цих елементів поступають на помножувач $M2$, що обчислює добуток ваги і відповідного йому коефіцієнта. Адреса ваги в ОЗП ваг визначається алгоритмом хешування, вхідним параметром якого є вектор асоціацій. Накопичуючий суматор $\Sigma 2$ спільно з помножувачем $M2$ обчислює (2.106) і після виконання ρ тактів у регістрі RgY формує значення функції, відповідне вхідним змінним. За допомогою суматора $\Sigma 3$ формується код помилки e_k , який визначається як різниця набутого значення функції u і потрібного u^* значень функції. Значення помилки поступає на блок реалізації алгоритму навчання, який під час ρ тактів обчислення накопичував адреси вибірки ваг, значення ваг і добуток функцій активації. Набувши значення помилки, алгоритм навчання модифікує ті, що брали участь в обчисленні ваги і записує їх в ОЗП ваг за відповідними адресами. Це відбувається під час обчислення значення функції для наступної комбінації вхідних змінних. Пам'ять має підтримувати одночасне читання і запис за різними адресами. Крім того, необхідно враховувати ситуацію, коли для поточної комбінації вхідних змінних необхідна вага, яка була модифікована під час навчання на попередній комбінації. Координацію роботи вузлів мережі здійснює пристрій управління.

2.5.2. Апаратна реалізація мережі СМАС у разі послідовного перебору значень змінних

Головним недоліком розглянутого способу апаратної реалізації нейронної мережі СМАС є його застосовність лише для значень ρ , рівних степеням числа 2. Причина в необхідності виконання ресурсомісткої операції ділення. Проте для оптимальної апроксимації функцій складного вигляду, якими є функції

присутності, а також самі зображення при їх стисненні, необхідно мати можливість задавати значення ρ , відмінні від степенів числа 2.

Розглянемо реалізацію мережі СМАС для завдань синтезу функцій присутності і стиснення зображень. Відмітною особливістю таких завдань є той факт, що значення змінних, відповідних координатам точок зображення на екрані, перебираються послідовно. Це відбувається унаслідок відрядкового розгортання зображення на екрані. Колір зображення може задаватися в різних колориметричних системах і визначається сукупністю складових (червоною, зеленою і синьою в системі RGB, яскравістю і кольорорізницевими складовими в системі YUV). Ці складові, у свою чергу, в загальному вигляді можуть подаватися у вигляді функцій $I(x,y)$ від координат точки на екрані x і y . При цьому координата x задається у вигляді тактової частоти f виведення точок в рядку, а координата y – у вигляді рядкових синхронізуючих імпульсів H-Sync. Функція присутності має такий же вигляд і розгортається аналогічно зображенню.

Розглянемо процес кодування змінних при їх послідовному переборі. У основі міркувань лежить той факт, що

$$\begin{aligned} x_{k,p} &= k, \\ y_{k,p} &= p. \end{aligned} \tag{2.107}$$

У зв'язку з тим, що дана функція залежить від двох змінних, приймемо двоіндексне позначення комбінацій цих змінних. При цьому індекс k відповідає за зміну змінних уздовж рядка з номером p . Метою аналізу є виявлення закономірностей при кодуванні змінних і визначенні змінної, для якої вибір номера ступеня, рівного номера кроку кодування, дасть кращий результат з огляду мінімуму апаратних витрат.

Обчислимо значення $u_x^{j_x}, NNS_x^{j_x}$ і $u_y^{j_y}, NNS_y^{j_y}$ відповідно до (2.100), (2.101) на різних кроках кодування для різних значень k і p . Результати зведемо

в таблицю 2.8 для випадків рівності номера кроку кодування і номера ступеня кодування для змінної x і y . Прийmemo $\rho = 5$.

Таблиця 2.8 – Розрахунок значень $u_x^{j_x}$, $NNS_x^{j_x}$ і $u_y^{j_y}$, $NNS_y^{j_y}$

(k,p)	j	$x_{(k,p)}$			$y_{(k,p)}$			$x_{(k,p)}$			$y_{(k,p)}$		
		j_x	$u_x^{j_x}$	$NNS_x^{j_x}$	j_y	$u_y^{j_y}$	$NNS_y^{j_y}$	j_x	$u_x^{j_x}$	$NNS_x^{j_x}$	j_y	$u_y^{j_y}$	$NNS_y^{j_y}$
(0,0)	0	0	0	0	1	1	0	1	1	0	0	0	0
	1	1	1	0	4	4	0	4	4	0	1	1	0
	2	2	2	0	2	2	0	2	2	0	2	2	0
	3	3	3	0	0	0	0	0	0	0	3	3	0
	4	4	4	0	3	3	0	3	3	0	4	4	0
(1,0)	0	0	1	0	1	1	0	1	2	0	0	0	0
	1	1	2	0	4	4	0	4	0	1	1	1	0
	2	2	3	0	2	2	0	2	3	0	2	2	0
	3	3	4	0	0	0	0	0	1	0	3	3	0
	4	4	0	1	3	3	0	3	4	0	4	4	0
(2,0)	0	0	2	0	1	1	0	1	3	0	0	0	0
	1	1	3	0	4	4	0	4	1	1	1	1	0
	2	2	4	0	2	2	0	2	4	0	2	2	0
	3	3	0	1	0	0	0	0	2	0	3	3	0
	4	4	1	1	3	3	0	3	0	1	4	4	0
(3,0)	0	0	3	0	1	1	0	1	4	0	0	0	0
	1	1	4	0	4	4	0	4	2	1	1	1	0
	2	2	0	1	2	2	0	2	0	1	2	2	0
	3	3	1	1	0	0	0	0	3	0	3	3	0
	4	4	2	1	3	3	0	3	1	1	4	4	0
(4,0)	0	0	4	0	1	1	0	1	0	1	0	0	0
	1	1	0	1	4	4	0	4	3	1	1	1	0
	2	2	1	1	2	2	0	2	1	1	2	2	0
	3	3	2	1	0	0	0	0	4	0	3	3	0
	4	4	3	1	3	3	0	3	2	1	4	4	0
(5,0) (k = ρ)	0	0	0	1	1	1	0	1	1	1	0	0	0
	1	1	1	1	4	4	0	4	4	1	1	1	0
	2	2	2	1	2	2	0	2	2	1	2	2	0
	3	3	3	1	0	0	0	0	0	1	3	3	0
	4	4	4	1	3	3	0	3	3	1	4	4	0
(6,0)	0	0	1	1	1	1	0	1	2	1	0	0	0
	1	1	2	1	4	4	0	4	0	2	1	1	0
	2	2	3	1	2	2	0	2	3	1	2	2	0
	3	3	4	1	0	0	0	0	1	1	3	3	0
	4	4	0	2	3	3	0	3	4	1	4	4	0
...													
(9,0)	0	0	4	1	1	1	0	1	1	2	0	0	0
	1	1	0	2	4	4	0	4	4	1	1	1	0
	2	2	1	2	2	2	0	2	2	2	2	2	0
	3	3	2	2	0	0	0	0	0	2	3	3	0

(k,p)	j	$x_{(k,p)}$			$y_{(k,p)}$			$x_{(k,p)}$			$y_{(k,p)}$		
		j_x	$u_x^{j_x}$	$NNS_x^{j_x}$	j_y	$u_y^{j_y}$	$NNS_y^{j_y}$	j_x	$u_x^{j_x}$	$NNS_x^{j_x}$	j_y	$u_y^{j_y}$	$NNS_y^{j_y}$
(10,10)	0	0	0	2	1	1	2	1	1	2	0	0	2
(k = 2ρ,	1	1	1	2	4	4	2	4	4	2	1	1	2
p = 2ρ)	2	2	2	2	2	2	2	2	2	2	2	2	2
	3	3	3	2	0	0	2	0	0	2	3	3	2
	4	4	4	2	3	3	2	3	3	2	4	4	2
...													

З таблиці 2.8 для випадку рівності номерів кроку квантування і ступеня для змінної x маємо такі закономірності:

– $u_x^{j_x+1}(k, p) = (u_x^{j_x}(k, p) + 1) \bmod \rho$, що з апаратної точки зору відповідає лічильнику CT_x^1 з модулем рахунку, рівним ρ , інкремент якого відбувається за кожним імпульсом тактової частоти кодування f_k ;

– $u_x^0(k+1, p) = (u_x^{\rho-1}(k, p) + 2) \bmod \rho$, що може реалізуватися шляхом введення додаткового такту рахунку для лічильника CT_x^1 для кожного значення k , що у свою чергу, викличе необхідність синтезу частоти кодування $(\rho+1) \cdot f$ за допомогою системи ФАПЧ;

– $NNS_x^{j_x+1}(k, p) = NNS_x^{j_x}(k, p) + (u_x^{j_x}(k, p) + 1) \operatorname{div} \rho$, що відповідає лічильнику CT_x^2 , інкремент якого відбувається за сигналом переповнювання лічильника CT_x^1 ;

$$- NNS_x^0(k+1, p) = \begin{cases} NNS_x^{\rho-1}(k, p) - 1, & k+1 \neq s \cdot \rho, s \in \text{int} \\ NNS_x^{\rho-1}(k, p), & k+1 = s \cdot \rho, s \in \text{int} \end{cases}$$

що може реалізуватися шляхом декремента лічильника CT_x^2 у додатковому такті рахунку, якщо в цьому такті відсутнє переповнювання лічильника CT_x^1 ;

– $u_x^{j_x}(k, p) = u_x^{j_x}(k, p+1)$ і $NNS_x^{j_x}(k, p) = NNS_x^{j_x}(k, p+1)$ – тобто немає залежності від номера рядка;

– $u_x^0(0, p) = 0$, $NNS_x^0(0, p) = 0$, отже, лічильники CT_x^1 і CT_x^2 повинні мати вхід скидання по рядковому синхронізуючому імпульсу H-Sync;

– $u_y^{j_y}(k, p) = (u_y^0(k, p) + j_y) \bmod \rho$ – відповідає двійковому суматору SM_y^1 з корекцією результату за модулем ρ . Цей блок може працювати відповідно до

$$\text{правила } u_y^{j_y}(k, p) = \begin{cases} (u_y^0(k, p) + j_y) - \rho, & (u_y^0(k, p) + j_y) - \rho < 0; \\ (u_y^0(k, p) + j_y) & (u_y^0(k, p) + j_y) - \rho \geq 0; \end{cases}$$

– $u_y^0(k, p) = u_y^0(k+1, p) = u_y^0(0, p)$ та $NNS_y^0(k, p) = NNS_y^0(k+1, p) = NNS_y^0(0, p)$, що означає, що значення $u_y^0(0, p)$ і $NNS_y^0(0, p)$ можуть розраховуватися один раз для всього рядка;

– $u_y^0(k, p+1) = (u_y^0(k, p) + 1) \bmod \rho$, що може реалізовуватися на лічильнику CT_y^1 з модулем рахунку, рівним ρ , інкремент якого відбувається за кожним рядковим синхронізуючим імпульсом H-Sync;

– $NNS_y^{j_y}(k, p) = NNS_y^0(k, p) + (u_y^0(k, p) + j_y) \operatorname{div} \rho$, що реалізується шляхом додавання 1 до значення $NNS_y^0(k, p)$ на даному кроці кодування, якщо на ньому є перенесення в суматорі SM_y^1 ;

– $NNS_y^0(0, p+1) = NNS_y^0(0, p) + (u_y^0(0, p) + 1) \operatorname{div} \rho$, що реалізується на лічильнику CT_y^2 шляхом його інкремента по рядковому синхронізуючому імпульсу H-Sync у разі наявності перенесення в лічильнику CT_y^1 ;

Отже, апаратні витрати на реалізацію блока кодування обох змінних складуть:

– лічильник CT_x^1 з кількістю розрядів, входами $m_x^1 = \lceil \log_2(\rho + 1) \rceil$ для інкремента і скидання та виходу переповнювання (для формування сигналу переповнювання необхідно $m_x^1/4$ комірок ПЛІС для випадку фіксованого ρ і $m_x^1/2$ – для випадку того, що змінюється ρ). Тут і далі вираз $\lceil * \rceil$ означає округлення до найближчого більшого цілого значення;

– лічильник CT_x^2 з кількістю розрядів $m_x^2 = \lceil \log_2(x_{\max} \operatorname{div} \rho) \rceil$ і входами для інкремента, декремента і скидання;

– суматор SM_y^1 з кількістю комірок LUT $3 \cdot m_y^1 = \lceil \log_2(\rho) \rceil$ (m_y^1 – для реалізації $(u_y^0(k, p) + j_y)$, m_y^1 – для реалізації, $(u_y^0(k, p) + j_y) - \rho$, m_y^1 – для вибору необхідного значення);

– лічильник CT_y^1 з кількістю розрядів $m_y^1 = \lceil \log_2(\rho) \rceil$, входами для інкремента і скидання та виходом переповнювання;

– лічильник CT_y^2 з кількістю розрядів $m_y^2 = \lceil \log_2(y_{\max} \text{ div } \rho) \rceil$ і входами для інкремента, декремента і скидання;

– система ФАПЧ, що синтезує частоту $f_k = (\rho + 1) \cdot f$.

Аналогічний аналіз проведемо для випадку рівності номера кроку кодування номера ступеня кодування для змінної u :

– $u_x^{j_x}(k, p) = (u_x^0(k, p) + j_x) \bmod \rho$ – що відповідає двійковому суматору SM_x^1 з корекцією результату за модулем ρ . Цей блок може працювати відповідно до правила

$$u_x^{j_x}(k, p) = \begin{cases} (u_x^0(k, p) + j_x) - \rho, & (u_x^0(k, p) + j_x) - \rho \geq 0; \\ (u_x^0(k, p) + j_x) & (u_x^0(k, p) + j_x) - \rho < 0 \end{cases};$$

– $u_x^0(k + 1, p) = (u_x^0(k, p) + 1) \bmod \rho$ – інкремент з частотою f лічильника CT_x^1 , що зберігає значення $u_x^0(k, p)$;

– $NNS_x^{j_x}(k, p) = NNS_x^0(k, p) + (u_x^0(k, p) + j_x) \text{ div } \rho$, що реалізується шляхом додавання 1 до значення $NNS_x^0(k, p)$ на даному кроці кодування, якщо на ньому є перенесення в суматорі SM_x^1 ;

– $NNS_x^0(k + 1, p) = NNS_x^0(k, p) + (u_x^0(k, p) + 1) \text{ div } \rho$, що реалізується на лічильнику CT_x^2 шляхом його інкремента з частотою f у разі наявності перенесення в лічильнику CT_x^1 ;

– $u_x^{j_x}(k, p) = u_x^{j_x}(k, p + 1)$ и $NNS_x^{j_x}(k, p) = NNS_x^{j_x}(k, p + 1)$ – тобто немає залежності від номера рядка;

– $u_x^0(0, p) = 0$, $NNS_x^0(0, p) = 0$, отже, лічильники CT_x^1 і CT_x^2 повинні мати вхід скидання по рядковому синхронізуючому імпульсу H-Sync;

– $u_y^{j_y+1}(k, p) = (u_y^{j_y}(k, p) + 1) \bmod \rho$ – лічильник CT_y^1 з модулем рахунку, рівним ρ , інкремент якого відбувається за кожним імпульсом тактової частоти кодування f_k ;

– $u_y^0(k+1, p) = (u_y^{\rho-1}(k, p) + 1) \bmod \rho$, що автоматично реалізується лічильником CT_y^1 ;

– $u_y^0(0, p+1) = (u_y^{\rho-1}(x_{\max} - 1, p) + 1) \bmod \rho$, що реалізується додатковим інкрементом лічильника CT_y^1 на початку нового рядка;

– $NNS_y^{j_y+1}(k, p) = NNS_y^{j_y}(k, p) + (u_y^{j_y}(k, p) + 1) \operatorname{div} \rho$, що відповідає інкременту лічильника CT_y^2 , який відбувається за сигналом переповнювання лічильника CT_y^1 ;

– $NNS_y^0(k, p) = NNS_y^0(k+1, p) = NNS_y^0(0, p)$, що може бути отримане шляхом декременту лічильника CT_y^2 після останнього ступеня кодування, тобто за сигналом переповнювання лічильника кроків кодування. При цьому, якщо виникає переповнювання лічильника CT_y^1 , ні інкремент, ні декремент лічильника CT_y^2 не проводиться.

Як видно з наведених міркувань, апаратні витрати за виключенням декількох вентилів в обох розглянутих випадках однакові. Однак другий випадок не вимагає додаткового такту кодування, і, відповідно, дозволяє знизити необхідну для кодування частоту. У першому випадку також можна обійтися без введення додаткового такту кодування. Це потребує додаткового лічильника для зберігання значення u_x^0 , інкремент якого має відбуватися за кожним імпульсом частоти f . Крім того, значення, що зберігається цим лічильником, має з частотою f записуватися в лічильник CT_y^1 . Тобто збільшаться апаратні витрати.

Оскільки в цілому немає принципової різниці, для якої змінної вибирати рівність кроку кодування і номера ступеня, синтез структури блока кодування змінних проводимо для другого випадку.

Структура блока кодування змінних для випадку послідовного перебору значень змінних і рівності номера кроку кодування номеру ступеня кодування для змінної u наведена на рис. 2.52.

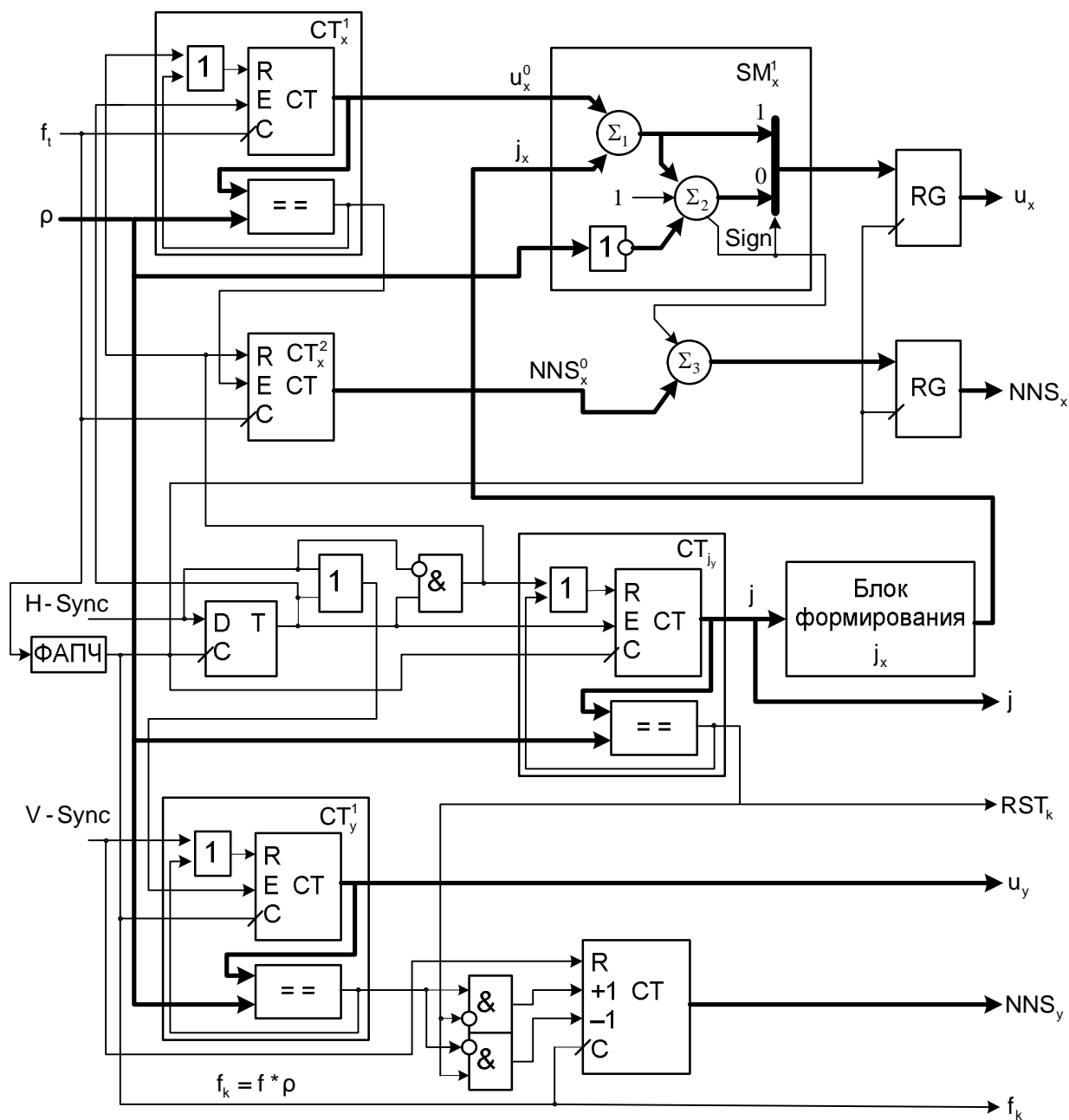


Рисунок 2.52 – Структура блока кодування змінних з послідовно перебираними значеннями

2.5.3 Стиснення і фільтрація зображень

Основною відмінністю застосування мережі СМАС для стиснення і фільтрації зображень є необхідність навчання в реальному масштабі часу. Це вимагає наявності засобів для проведення навчання, а також обумовлює необхідність заміни послідовного перебору змінних довільним, що, у свою чергу, викликає необхідність переробки кодуєчих блоків. Структура мережі СМАС, яка може використовуватися як самостійно, так і в ієрархічній структурі, наведено на рис. 2.53.

У пропонованій структурі для збереження можливості вибору довільних значень ρ у заданому інтервалі пропонується як кодуєчі блоки використовувати ОЗП відповідного розміру, що містять таблиці, за допомогою яких виконується перекодування змінних x і y в комбінації $u_x:NNS_x$, і $u_y:NNS_y$. Для стандартного телевізійного кадру розмірами 720 на 576 точок вхідні змінні подаються 10-розрядним двійковим кодом, а, отже, ОЗП для кодування мають містити по 1024 комірки, що мають розрядність 14 бітів для можливості задання ρ у діапазоні від 2-х до 32-х. Такі ОЗП можуть реалізовуватися в сучасних ПЛІС. Мультиплектори $MX1x$ і $MX1y$ і буферні елементи $B1x$ і $B1y$ необхідні для ініціалізації кодуєчих ОЗП.

Обчислення добутку базисних функцій здійснюється аналогічно розглянутим вище структурам. Ваги, що настроюються, зберігаються в ОЗП ваг, яке, на відміну від структур, що не вимагають навчання, має реалізовувати одночасне читання і запис за різними адресами.

Дана структура припускає навчання згідно з залежністю, яку запишемо нижче у вигляді, зручному для апаратної реалізації

$$w_j(k+1) = w_j(k) + \gamma \left(\frac{I_{in}(x(k), y(k)) - \sum_{i=0}^{\rho-1} \Psi_i(x(k))w_i(k)}{\sum_{n=0}^{\rho-1} (\Psi_n(x(k)))^2} \Psi_j(x(k)) \right). \quad (2.109)$$

Сума, що стоїть в чисельнику (2.109), є по суті значенням функції, що обчислюється мережею до настройки ваг. Вона обчислюється помножувачем $M1$ і накопичуючим суматором $\Sigma 1$. Після проходження ρ ступенів кодування в регістрі $RG1$ формується значення функції, обчислене мережею до навчання, яке за допомогою суматора $\Sigma 2$ віднімається з потрібного при даній вхідній комбінації змінних значення функції $I_{in}(x(k), y(k))$.

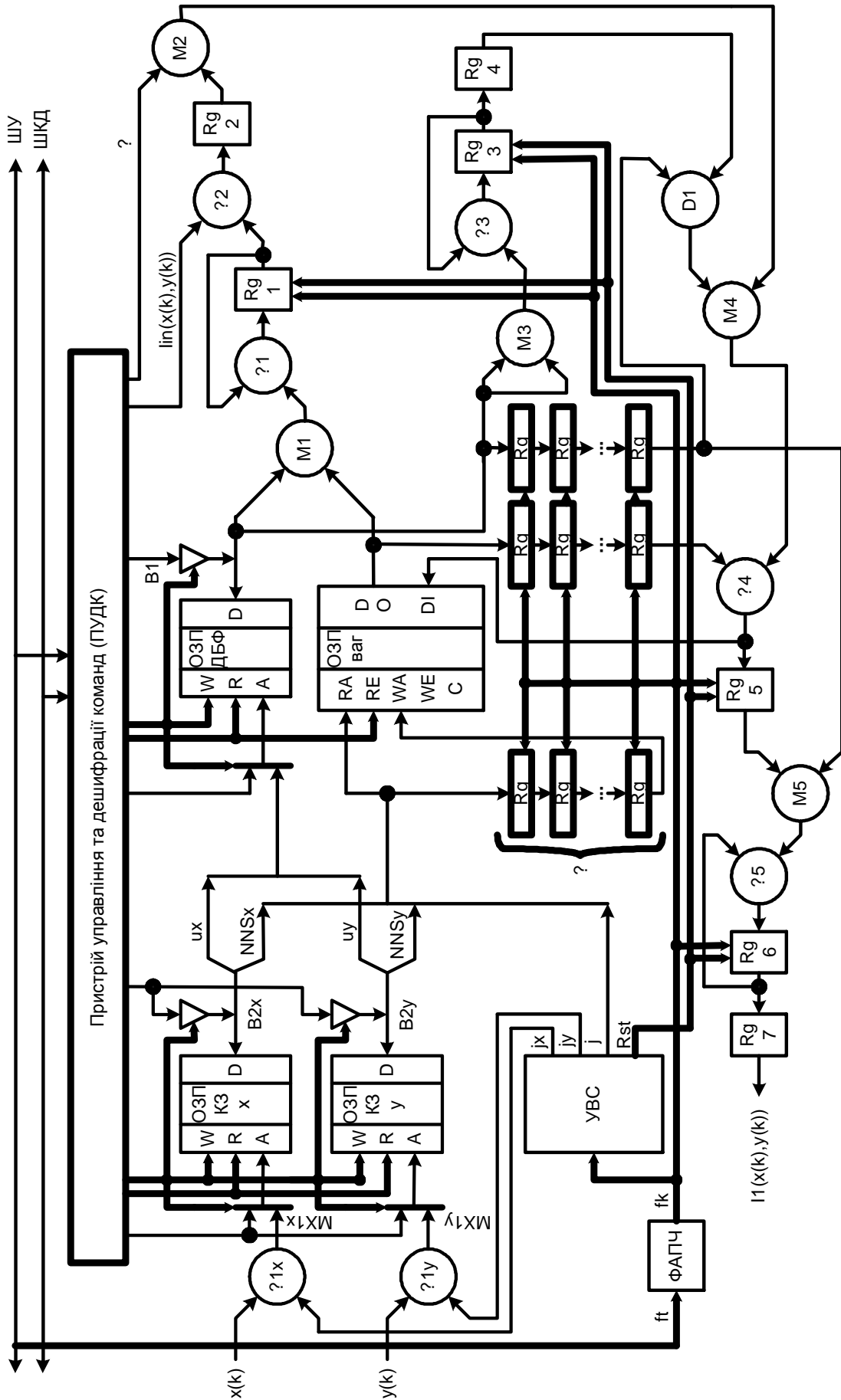


Рисунок 2.54 – Структура апаратної реалізації мережі СМАС, що вимагає навчання в реальному часі

Значення помилки зберігається в регістрі RG2, з якого поступає на помножувач M2, що обчислює добуток помилки на коефіцієнт γ , який визначає швидкість навчання мережі. Значення норми вектора коефіцієнтів, що стоїть в знаменнику (2.109) Ψ обчислюється помножувачем M3 і накопичуючим суматором $\Sigma 3$ -RG3 паралельно із значенням вихідної функції до навчання. Обчислене значення норми після завершення ρ ступенів кодування зберігається в регістрі RG5. Вектор коефіцієнтів, використовуваний для розрахунку значення функції до навчання, зберігається для подальшого використання в процесі навчання в ланцюжку зсувних регістрів. Аналогічно зберігаються адреси і значення ваг. Ці ваги будуть модифіковані в процесі навчання і записані в ОЗП за відповідними адресами. Довжина ланцюжків зсувних регістрів рівна ρ . Ці ланцюжки можуть реалізовуватися на вбудованих в ПЛІС блоках пам'яті, що включаються в двопортовий режим. Зміщення між адресами запису і читання має бути рівне ρ . Після затримки на ρ тактів, елементи вектора коефіцієнтів нормуються шляхом ділення на значення норми вектора коефіцієнтів в дільнику D1. Нормовані значення елементів вектора коефіцієнтів поступають на помножувач M4 спільно із значенням добутку помилки на коефіцієнт γ , який обчислює значення, що коректує, для ваги, яка модифікується на даному кроці навчання. Це значення, поступаючи на суматор $\Sigma 4$, додається до початкового значення ваги. Модифіковане значення ваги зберігається в регістрі RG5. З регістра RG5 спільно з власною адресою це значення поступає на лінії запису ОЗП ваг. Крім того, це значення поступає на помножувач M5 і накопичуючий суматор $\Sigma 5$ -RG6, функції, що обчислюють значення, після навчання мережі. Це значення зберігається в регістрі RG7 і надалі може використовуватися як результат фільтрації початкового зображення, або для навчання наступної в ієрархічному ланцюжку мережі.

Необхідно зазначити збільшення складності апаратної реалізації мережі СМАС за необхідності її навчання в реальному масштабі часу. Це пов'язано із збільшеною кількістю операцій множення і появою операції ділення при розрахунку величин корекції ваг. Існують алгоритми спільного виконання операцій

множення і ділення, проте у зв'язку з їх ітераційним характером вони в даній структурі непридатні. Потрібна їх адаптація до необхідної швидкості обчислень. Ці питання виходять за рамки даної дисертаційної роботи. Безпосереднє виконання операцій множення і ділення можливо на ПЛІС, що мають в своєму складі засоби цифрової обробки сигналів. Такі ПЛІС є на сучасному ринку, хоча все ще мають високу ціну. Внаслідок того, що кодуючої апаратури потрібно значно менше, ніж декодуючої, це не є серйозним недоліком.

При фільтрації зображень можливе навчання мережі на частини точок початкового зображення, що дозволяє знизити швидкість виконання обчислень поправок ваг і використовувати вищезазначені ітераційні алгоритми.

Мережі СМАС у декодуючій частині розглянутої ієрархічної структури не вимагають навчання і мають структуру, подібну наведеній на рис. 2.24. У цих мережах вхідні змінні перебираються послідовно, оскільки розгортка зображення відбувається в правильному порядку. Ці мережі ініціалізувалися вагами, що поступили каналом зв'язку і базисними функціями того ж вигляду, що використовуються на кодуючій стороні.

Наведемо порівняльну таблицю об'ємів пам'яті, необхідних для зберігання початкового зображення і стислого з використанням однієї мережі СМАС (табл. 2.9).

Таблиця 2.9 – Об'єми зображень після стиснення

Нестисле зображення розмірами 720·576 точок	Стисле з використанням мережі СМАС зображення, комірок			
	$\rho = 4$	$\rho = 8$	$\rho = 16$	$\rho = 32$
414720	104655	52981	27150	14246

З таблиці 2.9 видно, що скорочення об'єму інформації залежно від параметра мережі ρ може складати від 3,9 до 29,1 разів. За необхідності підвищення якості стислого зображення необхідно застосовувати ієрархічну структуру. В цьому випадку об'єм зображення дорівнюватиме сумі об'ємів пам'яті ваг, використовуваних у мережах ієрархічної структури.

При розгляді реалізації колірної корекції, синтезу функцій присутності, стиснення і фільтрації зображень не проводиться аналіз отримуваної похибки обчислень, яка залежить як від властивостей самої мережі СМАС і вибору її параметрів, так і від кількості розрядів при обчисленнях, вироблюваних у розглянутих структурах.

У зв'язку з тим, що пропонувані структури мають можливості вибору оптимального співвідношення (об'єм інформації)/(якість) і використовуються в області обробки відеоінформації, де критерієм якості є суб'єктивна оцінка, пропонується залишити за користувачем настройку параметрів структур через деякий інтерфейс і, відповідно, візуальну оцінку якості обробки.

2.6. Висновки до розділу 2

Наукова новизна та значимість отриманих наукових результатів полягає у наступному: розробка методів попередньої обробки зображень у системах технічного зору робототехнічних об'єктів; створення алгоритмів та програм для телевізійних систем технічного зору для розпізнавання й класифікації конкретних об'єктів, у т.ч. номерів автомобілів, порошкоподібних сумішей цементного клінкера; проведення експериментальних досліджень точності виконання нелінійних перетворень телевізійних зображень, швидкості виконання застосованих методів; введення корекції в досліджені алгоритми та методи для підвищення точності та швидкодії методів; перевірка виконання колориметричних перетворень на імітаційній моделі; математичне забезпечення та інструментальні засоби надшвидких перетворень, нормалізації і розпізнавання конкретної відеоінформації.

Створення на ПК діючих програмних моделей системи технічного зору для розпізнавання й класифікації конкретних об'єктів (у т.ч. номерів автомобілів, порошкоподібної суміші цементного клінкера). Програмні моделі повинні бути реалізованими на сучасних цифрових швидкодіючих апаратних засобах. Необхідно провести аналіз та дослідження існуючих методів, виявити їх недоліки, розробити методи та архітектурні принципи швидких геометричних перетво-

рювань зображень, а саме методи та алгоритми проєкцій нормалізації й розпізнавання зображень конкретних об'єктів. Розробити методи нелінійних алгоритмів колориметричної обробки зображень.

Дослідно-промислові зразки гнучких апаратно програмних цифрових систем технічного зору для розпізнавання й класифікації конкретних об'єктів, у т.ч. порошкоподібних сумішей цементного клінкера. Структури перетворювача координат та процесора колориметричної нормалізації зображення.

РОЗДІЛ 3. РОЗРОБКА ТЕОРІЇ ТА ПРИНЦИПІВ ПОБУДОВИ МОЗКОПОДІБНИХ ЕОМ З РЕЛЯЦІЙНИМИ МЕРЕЖАМИ

3.1 Загальна характеристика дослідження

Метою проекту є підвищення обробки інформації за рахунок розробки теорії та принципів побудови мозкоподібних ЕОМ з реляційними мережами.

Завдання проекту полягають в подальшій розробці інструментарію алгебри предикатів, методів опису інформаційних об'єктів засобами алгебри предикатів, методів побудови реляційних мереж мозкоподібних ЕОМ, зокрема, загального методу побудови реляційних мереж для довільних відношень:

- розробка інструментарію алгебри предикатів;
- розробка методів опису інформаційних об'єктів засобами алгебри предикатів;
- розробка методів побудови реляційних мереж мозкоподібних ЕОМ.

Основна ідея проекту полягає в розробці та впровадженні методів, що ґрунтуються на алгебро-логічному інструментарії алгебри предикатів як основного засобу реалізації відношень довільної природи.

Можливості розпаралелювання обробки інформації на алгоритмічному шляху суттєво обмежені порівняно з реляційним шляхом. Тому нові методи направлені саме на розпаралелювання, а саме на створення методів та моделей, що дозволяють будувати реляційні мережі паралельної дії та апаратно їх реалізувати. Запропоновані методи та підходи базуються на засадах теорії графів, алгебри предикатів, теорії категорій та реляційної алгебри.

Об'єктом дослідження є паралельна обробка інформації у природі і технічних пристроях. Предмет дослідження – реляційна обробка інформації в природних об'єктах та її технічна реалізація в системах штучного інтелекту.

3.2. Опис процесу наукового дослідження

Реляційній обробці протистоить алгоритмічна (функціональна) її обробка. З самого початку розвитку ЕОМ в них реалізовано алгоритмічну обробку інфо-

рмації, для чого було використано покрокове програмування. До цього часу зусилля фахівців були спрямовані на розпаралелювання машинної обробки інформації виключно в рамках алгоритмічного принципу дії ЕОМ. Відтворення в ЕОМ реляційного принципу обробки інформації дасть змогу суттєво підвищити продуктивність ЕОМ і розширити спектр задач, які буде можливо ефективно вирішувати за допомогою комп'ютера у реальному часі.

Процес пошуку більш досконалих технічних засобів, здатних вирішувати рівняння алгебри скінченних предикатів (АСП), привів до реляційних мереж. Ці технічні засоби орієнтовані на паралельну обробку інформації; вони показали свою простоту та ефективність. Для реалізації логічних мереж добре підходять недорогі інтегральні схеми програмувальної логіки. Наявні моделі природної мови, представлені рівняннями АСП, об'єднано в єдину модель і реалізовано у вигляді реляційної мережі. Така мережа може стати апаратною частиною систем машинного перекладу, систем розпізнавання тексту й мови, та інших систем, що вирішують задачі автоматичної обробки природної мови.

Модель, придатна до реалізації у вигляді реляційної мережі, представлена системою бінарних рівнянь АСП (тобто кожне рівняння такої системи повинне мати дві змінні). Така форма подання є моделлю логічної мережі; при цьому система бінарних рівнянь повинна бути як можна більше компактною, щоб заощадити ресурси апаратури при реалізації моделі.

Тим часом теорія реляційних баз даних пропонує засоби декомпозиції відношень у вигляді тверджень про залежності між атрибутами (у розділі, присвяченому нормалізації відношень). З огляду на глибокий зв'язок між реляційними відношеннями і предикатами, можна звернутися за засобами декомпозиції відношень для рішення задачі декомпозиції предикатів. Як показав аналіз літератури, такі засоби в апараті алгебри предикатів на даний момент відсутні.

Перелічені обставини переконали авторів проекту у тому, що алгебра предикатів є саме тим інструментом, яким природа скористалась, синтезуючи нейронні мережі мозку людини. Це дало підставу рекомендувати реляційний

шлях розвитку обчислювальної техніки, як нову перспективну можливість її подальшого удосконалення.

3.3. Отримані найважливіші наукові та науково-технічні результати

Реляційна обробка інформації широко розповсюджена у природі. Більшість процесів у зовнішньому світі можна розглядати як рішення рівнянь теорії поля. На реляційному принципі працюють незаслужено забуті цифрові диференціальні аналізатори. Відомо, що основна частина інформації обробляється людиною на підсвідомому рівні за допомогою асоціативних мереж. На долю свідомої (алгоритмічної) обробки інформації людиною припадає мізерний процент її обсягу. Реляційний принцип обробки інформації реалізовано природою в діяльності мозку людини.

Для обробки й повноцінного моделювання природної мови в реальному режимі часу потрібна паралельна апаратна платформа. У ролі архітектури для такої платформи кілька років назад були запропоновані логічні мережі. Проте необхідно заповнити недолік математичних засобів, необхідних для побудови моделей логічних мереж, і продемонструвати ефективність нових засобів. Для досягнення цієї мети у вступі до роботи був поставлений ряд задач. Очікуються наступні результати:

- аналіз сучасного стану проблем в області розробки логічних мереж і математичних засобів для побудови їх моделей. Виявлення недоліків засобів у цій області. Обґрунтування розробки засобів кон'юнктивної декомпозиції, що базуються на теорії нормалізації реляційних відношень і орієнтованих на подання предикатів у бінарній формі (придатної для побудови моделей реляційних мереж);

- удосконалення формального опису зв'язку між операціями в реляційній алгебрі й в алгебрі скінченних предикатів. Завдяки цьому будь-які вирази реляційної алгебри буде можливим перетворити у вирази алгебри предикатів;

- подальший розвиток алгебро-логічного апарату декомпозиції предикатів: деякі реляційні твердження про залежності переформульовані в термінах

алгебри предикатів за допомогою теорії нормалізації реляційних відношень; розроблені нові критерії перевірки предикатів на виконання залежності кон'юнкції певного виду;

- розробка та обґрунтування методу бінарної декомпозиції функціональних предикатів, що відрізняється від загального методу декартової декомпозиції тим, що кількість значень допоміжної змінної зведено до мінімуму;

- побудова моделі реляційної мережі для флективної обробки дієслівних слів форм російської мови, та її апаратна реалізація у вигляді IP-core (модуля, захищеного від несанкціонованого доступу, готового до імплементації в кристал програмувальної логіки). Оцінка розмірів моделі.

- побудова математичної моделі відмінювання словосполучень, на основі існуючих моделей відмінювання прикметників та іменників;

- розробка методу побудови реляційної мережі для виконання арифметичних дій (додавання та множення) десяткових чисел заданого розміру та його програмна реалізація.

Розроблено спеціалізований обчислювальний комплекс, за допомогою якого здійснюється автоматичне проектування і виготовлення мозкоподібної ЕОМ, що реалізує модель довільно обраного механізму мови і мислення людини. Поки мозкоподібна ЕОМ реалізує модель дуже невеликого механізму природної мови, що відає відмінюванням повних неприсвійних імен прикметників. Для цього був потрібен один ЧІП, вартістю 10 доларів, складність якого вимірюється 20 тисячами вентилів. Мозкоподібна ЕОМ вирішує цю задачу за десятку частку мікросекунди, людина – за десятку частку секунди. Яка ж частина функцій мозку людини реалізована сьогодні мозкоподібною ЕОМ? Поділяємо число $2 \cdot 10^4$ вентилів ЧІПа на число 10^{15} усіх "вирішальних елементів" мозку. Одержуємо $0,2 \cdot 10^{-10}$. Це складає дві мільярдних частки відсотка! Які перспективи підняти величину цього відсотка? Перше – це матеріальна сторона справи. Уже зараз розроблювачі мозгоподібної ЕОМ розташовують ЧІПами на 100 тисяч вентилів вартістю по 50 доларів кожний. Ніщо не заважає з'єднати них на

паралельну роботу. У продажі маються ЧПІ на 10 млн. вентилів: кожний з них коштує близько 10 тис. доларів, але їхня ціна швидко падає. У стадії розробки знаходяться ЧПІ, число вентилів яких на два порядки вище. Спостерігається також швидке зниження розмірів і ваги ЧПІв. Так що цілком реально в найближчі роки мати в розпорядженні мозкоподібну ЕОМ з числом вентилів порядку 10^8 . Складніше "прокормити" таку мозкоподібну ЕОМ. Для цього знадобиться $10^8:(2 \cdot 10^4)$, тобто п'ять тисяч моделей, подібної тієї, котра зараз реалізована на мозкоподібній ЕОМ. На черзі створення мозкоподібної ЕОМ, що самонавчається й самоудосконалюється. Алгебрологічні основи побудови такого типу мозкоподібної ЕОМ уже закладені. Гранична складність задачі, що зможе вирішити мозкоподібна ЕОМ, визначається лише числом вентилів у її процесорі. В даний час процесор мозкоподібної ЕОМ навчається за допомогою звичайного комп'ютера послідовної дії. Навчання зводиться до формування схеми з'єднань вирішальних елементів у ЧПІ, обумовленою моделлю реалізованого в мозкоподібній ЕОМ того або іншого механізму мови і мислення. Може бути логічні мережі мозкоподібних ЕОМ вдасться використовувати для протезування ушкоджених ділянок мозку людини.

3.4. Відмінні риси і перевага отриманих результатів (продукції) над вітчизняними або зарубіжними аналогами чи прототипами

Відомо, що окремі вирішуючі елементи нейронної мережі мозку (нейрони та синапси) працюють у 10^6 разів повільніше, ніж вирішуючі елементи сучасної ЕОМ. Проте продуктивність мозку людини при обробці складної інформації перевищує продуктивність найпотужніших сучасних ЕОМ у 10^6 разів. Цей факт пояснюється тим, що 10^{11} нейронів і 10^{15} синапсів мозку людини працюють паралельно і одночасно. Вирішуючі ж елементи процесора ЕОМ працюють послідовно, оскільки це обумовлено алгоритмічним принципом дії програм, керуючих дією процесора ЕОМ.

Практична реалізація принципу дії мозку людини безпосередньо веде до створення мозкоподібних ЕОМ із процесором у вигляді реляційної мережі.

Співавтори проекту Бондаренко М.Ф. і Шабанов-Кушнарченко Ю.П. розробили алгебру предикатів, за допомогою якої відкрили принцип дії нейронної мережі мозку людини, яка функціонує не алгоритмічно, а реляційно, вирішуючи рівняння алгебри предикатів. Практична реалізація принципу дії мозку людини безпосередньо веде до створення мозкоподібних ЕОМ із процесором у вигляді реляційної мережі. Ефективність дії мозкоподібної ЕОМ із реляційною мережею була продемонстрована на макеті, який створено на базі персонального комп'ютера під керівництвом д.т.н., проф. кафедри АПОТ ХНУРЕ Хаханова В.І.

Як вихідну інформацію при визначенні продуктивності мозкоподібних ЕОМ, була використана максимальна частота переключення електронних вирішальних елементів, застосовуваних у сучасних ЕОМ - 10^9 гц. Було також прийняте, що число одночасно працюючих вирішальних елементів у комп'ютерах послідовної дії у середньому не перевищує 10^3 . У результаті була отримана наступна оцінка максимальної продуктивності сучасної ЕОМ послідовної дії: $10^9 \cdot 10^3 = 10^{12}$ елементарних операцій на секунду. Біологами встановлено, що гранична частота проходження імпульсів по нервових волокнах мозку не перевищує 10^3 гц. Отже, "біологічні" вирішальні елементи не можуть переключатися з частотою вище 10^3 гц. Дані нейрофізіології свідчать, що мозок людини являє собою пристрій паралельної обробки інформації. У ролі "біологічних вирішальних елементів" мозку нейрофізіологи приймають синапси, загальне число яких у мозку людини оцінюється величиною 10^{15} . Спираючись на ці дані, приходимо до наступної оцінки продуктивності мозку людини: $10^3 \cdot 10^{15} = 10^{18}$ переключень у секунду, що перевершує продуктивність сучасної обчислювальної машини послідовної дії в $10^{18} : 10^{12} = 10^6$ разів. Були також початі спроби оцінити максимально досяжну продуктивність обчислювальної машини паралельної дії, що працює по принципах роботи мозку людини. Виходячи зі стійких темпів, що спостерігаються фактично, зменшення вартості, розмірів і ваги електронних вирішальних елементів і прогресу в області автоматизації проектування і виготов-

лення схем ЕОМ, був зроблений висновок про можливість створення в доступному для огляду майбутньому обчислювальних структур, по своїй складності рівноцінній складності нейронної мережі мозку людини. Мозкоподібна ЕОМ буде здатна в перспективі розвинути продуктивність до $10^9 \cdot 10^{15} = 10^{24}$ операцій у секунду. Це означає, що перехід до мозкоподібних ЕОМ забезпечить у доступному для огляду майбутньому підвищення продуктивності обчислювальних машин у порівнянні з нинішніми комп'ютерами послідовної дії в $10^{24} : 10^{12} = 10^{12}$ разів. За рахунок переходу від послідовних до паралельних методів обробки інформації вже сьогодні досягнуте підвищення продуктивності ЕОМ приблизно в 10^4 разів. Проте ще залишається великий резерв росту продуктивності ЕОМ у 10^8 разів.

3.5. Висновки до розділу 3

Результати роботи рекомендуються для розроблювачів ЕОМ і творців програм, що імітують на ЕОМ інтелектуальну діяльність людини. Товарна продукція має вигляд карт для персональних ЕОМ, що моделюють різні інтелектуальні процеси: програми машинного перекладу, удосконалення природномовного інтерфейсу, пошуку необхідної інформації в інтернеті, розпізнання об'єктів, машинне розуміння мови, створення інтелектуальних агентів різного призначення, удосконалення баз знань, методів подання знань тощо.

Принципових перешкод на шляху створення серійно діючих мозкоподібних ЕОМ немає. При відповідному фінансуванні і організації науково-дослідних і практичних робіт ця задача може бути вирішена за 5-10 років. Продуктивність ЕОМ при цьому може бути підвищена у 10^6 разів, а при повній реалізації принципу дії мозку в ЕОМ – у 10^{12} разів. Досягнення цієї мети призведе до того, що Україна стане одним із світових лідерів у галузі виробництва надпотужних ЕОМ паралельної дії.

В учбовому процесі планується застосування результатів у наступних курсах: «Основы теории искусственного интеллекта», «Теория интеллекта», «Модели языка», «Логический анализ», «Методологические основы научных

исследований». Результати досліджень по цьому напрямку представлені в науково-технічному журналі «Бионика интеллекта». Будуть розроблені та зареєстровані у фонді алгоритмів університету окремі програмні продукти. Також планується підготовка та захист кандидатських дисертацій на тему: «Математичні моделі суб'єктивного оцінювання та їх застосування в соціально-економічних системах», «Кількісні оцінки фоносемантичних характеристик текстів природної мови та їх використання в системах штучного інтелекту» та докторських на тему: «Методи, моделі та алгоритми вилучення знань з текстових баз даних», «Методи синтезу надшвидкодіючих структур та їх застосування для автоматизованої обробки українських текстів», «Математичні основи теорії факторизації на базі довільних відношень та її додатків».

РОЗДІЛ 4. ІНШІ РЕЗУЛЬТАТИ, ОТРИМАНІ ТА РЕАЛІЗОВАНІ В РАМКАХ НДР

4.1. Монографії, підручники та навчальні посібники

Монографії:

1. Хаханов В.И., Литвинова Е.И., Гузь О.А. Проектирование и тестирование цифровых систем на кристаллах (монографія). Харьков: ХНУРЕ.– 2009. 484 с. (48,4)

2. Хаханов В.І. Інфраструктура мозгоподібних обчислювальних процесів / Хаханов В.І., Бондаренко М.Ф., Шабанов-Кушнарєнко Ю.П., Гузь О.О. – Харків. – 2010. – 160 с.

3. М.Ф. Бондаренко, Ю.П. Шабанов-Кушнарєнко. – Мозгоподобные структуры: Справочное пособие. Том первый // Под редакцией акад. НАН Украины И.В. Сергиенко. – К.: Наукова думка, 2011. – 460с.

4. Сліпченко М.І., Руденко О.Г., Сотников О.М., та ін. Проблемно-орієнтовані і спеціалізовані обчислювальні засоби високої продуктивності: створення та застосування /За заг. ред. М.Ф.Бондарєнка. – Харків: ХНУРЕ, 2011. . – 492 с.

5. Єлаков С.Г., Зінченко О.О., Сліпченко М.І. та ін. Теорія та методи створення проблемно-орієнтованих обчислювальних засобів високої продуктивності / За заг. ред. М.Ф.Бондарєнка. – Харків: ХНУРЕ, 2010. – 400 с.

6. Зінченко О.О., Сліпченко М.І., Єлаков С.Г., Капран І.І., Лісовенко О.П., Новиков М.В., Руденко О.Г. Сотников О.М. Методи та спеціалізовані обчислювальні засоби швидкої обробки великих обсягів інформації. За загальною редакцією чл.-кор. НАНУ М.Ф. Бондарєнка. Україна, Харків, "Навчально-науковій видавничо-поліграфічній центр ХНУРЕ", 336 с.

7. Hahanov V. Infrastructure Intellectual Property for SoC Simulation and Diagnosis Service / Розділ в монографії «Design of Digital Systems and Devices» під редакцією М. Adamski, А. Barkalov, М. Wegrzyn. – Springer. – 2011. – P. 289-330.

Підручники:

1. Бондаренко М.Ф., Шубін І.Ю. Мультимедіа системи. - Харків: ХНУРЕ, 2010. – 300 с.
2. Шкиль А.С., Сыревич Е.Е., Кораблев Н.И. Учебное пособие по курсу «Компьютерная арифметика» для дистанционного обучения для специальности 6.091500 «Компьютерная инженерия» в формате Lersus. Сертификат № 015 от 24.06.09. 300 с.
3. Кривуля Г.Ф. Рябенський В.М., Рязанцев О.І. Комп'ютерна схемотехніка (підручник). Луганськ – 2009. СДУНУ ім. В. Даля. 744 с. (67,6)

Навчальні посібники:

1. Хаханов В.И. Verilog and System Verilog / В.И. Хаханов, Е.И. Литвинова, И.В. Хаханова, О.А. Гузь. – Харьков.– 2010.– 528 с.

4.2. Патенти

1. Бурцев Вал.М., Бурцев Вол.М., Єрохін А.Л., Стеценко О.П. Світловий інформаційний пристрій для управління дорожнім рухом / Патент на винахід №86672. МПК G08G 1/09, G09F 9/00. Зареєстровано 12.05.2009. Бюл.№9. – 34 с.
2. Рішення про видачу деклараційного патенту на корисну модель по заявці № u 2009 08123 від 03.08.2009, авторів Новиков Н.В., Єлаков С.Г., Сотников О.М. "Спосіб вимірювання температури поверхонь, що обмежують великі об'єми".
3. Патент на винахід №86635 від 12.05.2009 г. "Телевізійна система спостереження за рухомими об'єктами". Єлаков С.Г., Торба А.А., Мегель Ю.Є., Сотников О.М. та інш.

4.3. Статті

За результатами наукового дослідження у 2009-2011 роках підготовлено та опубліковано 114 статей у фахових виданнях:

1. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Идентификация объектов, описываемых векторами и линейными операциями над ними // Бионика интеллекта. – 2009. – № 1 (70) – С.3-10.

2. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Разработка систем условий, обеспечивающих существование линейного конечномерного оператора // Бионика интеллекта. – 2009. – № 1 (70) – С.11-16.

3. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Разработка теории структурной метризации конечномерного арифметического пространства // Бионика интеллекта. – 2009. – № 1 (70) – С.17-27.

4. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Практические приложения компараторной идентификации линейных конечномерных объектов // Бионика интеллекта. – 2009. – № 1 (71) – С.5-12.

5. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Разработка системы экономного кодирования цвета // Бионика интеллекта. – 2009. – № 1 (71) – С.13-23.

6. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Методы идентификации механизма субъективных социально-экономических оценок // Бионика интеллекта. – 2009. – № 1 (71) – С.24-30.

7. Bondarenko M. F., Chetverikov G. G., Vechirskaya I.D. Formal models of language systems // Вестник ХНАДУ: Сб. научн.труд. – 2009. – Вып. 35. – С. 115-117.

8. Вечирская И.Д., Четвериков Г.Г., Федорова Т.Н. Расслоение предикатов на примере словоизменения прилагательных русского языка // Искусственный интеллект. – 2009. – № 3 – С. 170 – 177.

9. Вечирская И.Д., Четвериков Г.Г., Валенда Н.А. Логическая сеть как средство представления естественных языковых знаний // Труды Международной научной конференции “Горизонты прикладной лингвистики и лингвистических техно-логий”. – Киев, Довіра, 2009.- С.251-257.

10. Вечирская И.Д., Четвериков Г.Г., Федорова Т.Н., Богданов Д.Ф. Моделирование фрагментів природної мови алгебро-логічними засобами (на прикладі k-значної реалізації морфологічного аналізу)// Бионика интеллекта. – 2009. – № 1 (70) – С. 55 – 59.

11. Четвериков Г.Г., Королева Я.Ю., Бережная М.А. Синтез проверяющих тестов для однородных схем // Бионика интеллекта. – 2009. – № 2 (71) – С. 123 – 127.
12. Вечирская И.Д. Розв'язання задачі знаходження гіпотетично зв'язаних об'єктів // Право і безпека. – 2009. – № 2 – С. 268-273.
13. Вечирская И.Д. Дослідження розмірності предметного простору в задачах моделювання об'єктів у вигляді реляційних мереж // Бионика интеллекта. – 2009. – № 2 (71) – С. 31 – 35
14. Ерохин А.Л., Кольченко А.В. Интеллектуальный контроль пользователей в информационных системах // Бионика интеллекта: научн.-техн. журнал. 2009. №1(70). - С.39-43.
15. Єрохін А.Л., Кольченко О.В. Интеллектуальна система підтримки взаємодії з комп'ютером для людей з руховими порушеннями // Біоніка інтелекту: научн.-техн. журнал. 2009. №2(71). – С.66-69.
16. Єрохін А. Л., Романов В. О. Розробка додатків засобами бібліотеки wxWidgets // Право і безпека. №2. - 2009. - С.250-255.
17. Єрохін А.Л., Турута О.П. Оцінка ефективності обслуговування користувача в інформаційній мережі // Право і безпека. №3. - 2009. - С.164-165.
18. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Модель равенства идей // Бионика интеллекта. – 2010. – № 2 (73) – С.3-15.
19. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Алгебра идей // Бионика интеллекта. – 2010. – № 2 (73) – С.16-27.
20. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Метод сравнения// Бионика интеллекта. – 2010. – № 2 (73) – С.28-39.
21. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Изоморфизмы алгебры идей// Бионика интеллекта. – 2010. – № 2 (73) – С.40-50.

22. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Интерпретация алгебры идей // Бионика интеллекта. – 2010. – № 2 (73) – С.51-61.

23. Бондаренко М.Ф., Кругликова Н.П., Лещинская И.А., Русакова Н.Е., Шабанов-Кушнарченко Ю.П. Об алгебре одноместных предикатов // Бионика интеллекта. – 2010. – № 2 (73) – С.62-67.

24. Бондаренко М.Ф., Русакова Н.Е., Шабанов-Кушнарченко Ю.П. О мозгоподобных структурах // Бионика интеллекта. – 2010. – № 2 (73) – С.68-73.

25. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шаронова Н.В. Инструментарий компараторной идентификации // Бионика интеллекта. – 2010. – № 2 (73) – С.74-86.

26. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шаронова Н.В. Ситуационно-текстовый предикат// Бионика интеллекта. – 2010. – № 2 (73) – С.87-98.

27. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шаронова Н.В. Булева структура текста // Бионика интеллекта. – 2010. – № 2 (73) – С.99-110.

28. Бондаренко М.Ф., Кругликова Н.П., Русакова Н.Е., Шабанов-Кушнарченко Ю.П. О методе нулевого прибора // Бионика интеллекта. – 2010. – № 2 (73) – С.111-115.

29. Бондаренко М.Ф., Кругликова Н.П., Пославский С.А., Шабанов-Кушнарченко Ю.П. О теории натурального ряда // Бионика интеллекта. – 2010. – № 2 (73) – С.129-139.

30. Бондаренко М.Ф., Кругликова Н.П., Пославский С.А., Шабанов-Кушнарченко Ю.П. О теории рациональных чисел // Бионика интеллекта. – 2010. – № 2 (73) – С.140-149.

31. Бондаренко М.Ф., Дрюк А.Д., Кругликова Н.П., Пославский С.А., Шабанов-Кушнарченко Ю.П. О теории действительных чисел // Бионика интеллекта. – 2010. – № 2 (73) – С.150-158.

32. Бондаренко М.Ф., Работягов А.В., Щепковский С.В. Распознавание речи: этапы развития, современные технологии и перспективы их применения // Бионика интеллекта. – 2010. – № 2 (73) – С.164-168.

33. Четвериков Г.Г. Концепція уніфікації методів та засобів побудови просторових багатозначних структур мовних систем // Бионика интеллекта. – 2010. – № 1 (72) – С.3-11.

34. Вечірська І.Д., Федорова Т.М., Четвериков Г.Г. Побудова функціональних моделей елементів комп'ютерної лінгвістики // Бионика интеллекта. – 2010. – № 1 (72) – С.83-88.

35. Русакова Н.Е. Модель устной речи // Бионика интеллекта. – 2010. – № 1 (72) – С.94-97.

36. Бондаренко М.Ф., Кругликова Н.П., Лещинская И.А., Русакова Н.Е., Шабанов-Кушнаренко Ю.П. Об алгебре предикатов // Бионика интеллекта. – 2010. – № 3 (74) – С.3-7.

37. Бондаренко М.Ф., Кругликова Н.П., Лещинская И.А., Русакова Н.Е., Шабанов-Кушнаренко Ю.П. О реляционных сетях // Бионика интеллекта. – 2010. – № 3 (74) – С.8-13.

38. Вечирская И.Д., Валенда Н.А. Логическое преобразование как средство реализации сетевых технологий // Бионика интеллекта. – 2010. – № 3 (74) – С.14-18.

39. Алексієв О.П., Алексієв В.О., Хабаров В.О., Четвериков Г.Г. Інформатизація транспортної інфраструктури, машин та систем // Бионика интеллекта. – 2010. – № 3 (74) – С.52-57.

40. Вечірська І. Д. Застосування методу декомпозиції предикатів для реалізації довільних відношень реляційними мережами // Право і безпека. – 2010. – № 1(33) – С. 237-241.

41. Четвериков Г.Г., Процай Н.Т. Аналіз функціональної стабільності інформаційних систем з інтелектуальним керуванням // Право і безпека. – 2010. – № 1(33) – С. 218-222.

42. Г.Четвериков, І.Вечірська, Т.Федорова, І.Лещинська. Концепція застосування алгебри предикатів для побудови просторових структур природної мови // Збірник наукових праць „Прикладна лінгвістика та лінгвістичні технології” – Київ, Довіра, 2009. – С.461 – 469.

43. Бондаренко М.Ф., Каменева І.В., Русакова Н.Е., Шабанов-Кушнарченко Ю.П., Шубин І.Ю. О булевых реляционных сетях // Бионика интеллекта. – 2011. – № 1 (75) – С.3-7.

44. Бондаренко М.Ф., Кругликова Н.П., Русакова Н.Е., Шабанов-Кушнарченко Ю.П. О логической идентификации объектов // Бионика интеллекта. – 2011. – № 1 (75) – С.8-12.

45. Бондаренко М.Ф., Русакова Н.Е., Шабанов-Кушнарченко Ю.П. О мозгоподобных структурах академика Виктора Михайловича Глушкова // Бионика интеллекта. – 2011. – № 2 (76) – С.3-9.

46. Бондаренко М.Ф., Русакова Н.Е., Шабанов-Кушнарченко Ю.П. О предикатной категории // Бионика интеллекта. – 2011. – № 2 (76) – С.10-23.

47. Бондаренко М.Ф., Кругликова Н.П., Русакова Н.Е., Шабанов-Кушнарченко Ю.П. Проблемы моделирования субъективных состояний // Бионика интеллекта. – 2011. – № 2 (76) – С.24-32.

48. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Линейные предикаты и их применение для моделирования цветового зрения человека // Бионика интеллекта. – 2011. – № 2 (76) – С.33-51.

49. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. О системе условий линейности предиката // Бионика интеллекта. – 2011. – № 2 (76) – С.52-64.

50. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П., Шабанов-Кушнарченко С.Ю. Интегральные представления линейных предикатов // Бионика интеллекта. – 2011. – № 2 (76) – С.65-78.

51. Бондаренко М.Ф., Шабанов-Кушнарченко С.Ю., Шабанов-Кушнарченко Ю.П. Дедуктивное построение теории цвета // Бионика интеллекта. – 2011. – № 2 (76) – С.79-85.

52. Бондаренко М.Ф., Шабанов-Кушнарченко С.Ю., Шабанов-Кушнарченко Ю.П. Модели компараторной идентификации в виде семейств одно- и двух-параметрических операторов // Бионика интеллекта. – 2011. – № 2 (76) – С.86-97.

53. Бондаренко М.Ф., Шабанов-Кушнарченко С.Ю., Шабанов-Кушнарченко Ю.П. Модели компараторной идентификации в виде семейств трех-параметрических и и сверточных операторов // Бионика интеллекта. – 2011. – № 2 (76) – С.98-108.

54. Вечирская И.Д. Разработка трехязычного терминологического словаря на основе алгебры конечных предикатов // Бионика интеллекта. – 2010. – № 2 (76) – С.109-113.

55. I. Vechirska, G. Chetverikov, V. Leshchynskiyi. Solving the problem of finding hypothetically connected objects by means of spatial multi-valued structures // Искусственный интеллект. – 2011. – № 3 – С. 319 – 326.

56. Четвериков Г., Вечірська І., Федорова Т. Структура элементов компьютерной лингвистики и пути ее моделирования // Труды Международной научной конференции “Горизонты прикладной лингвистики и лингвистических технологий”. – Киев, Довіра, 2010. – С. 206-214.

57. М.Ф. Бондаренко, И.А. Лещинская, Н.Е. Русакова, В.И. Хаханов. Инфраструктура анализа логических ассоциативных отношений // Радіоелектроніка і інформатика: наук.-техн. журнал. – 2010. – №1(48). – С. 38–49.

58. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П. Об алгебре конечных предикатов // Бионика интеллекта. – 2011. – № 3 (77) – С.3-13.

59. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П. Нормальные формы формул алгебры конечных предикатов // Бионика интеллекта. – 2011. – № 3 (77) – С.14-29.

60. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П. Уравнения теории интеллекта // Бионика интеллекта. – 2011. – № 3 (77) – С.30-45.

61. Голян Н.В. Шабанов-Кушнарченко Ю.П. Предикатные модели неявных связей между процедурами бизнес-процесса // Бионика интеллекта. – 2011. – № 3 (77) – С.46-49.

62. Русакова Н.Е. О методе расслоения конечного предиката // Бионика интеллекта. – 2011. – № 3 (77) – С.50-53.

63. Бондаренко М.Ф., Коноплянко З.Д., Четвериков Г.Г. Концепції уніфікації інформаційно -інтелектуальних технологій в системах мовлення // Бионика интеллекта. – 2011. – № 3 (77) – С.150-156.

64. Елаков С.Г., Новиков Н.В., Слипченко Н.И., Сотников О.М., Лисовенко А.П./ Оптимизация расхода электроэнергии при плавке в руднотермических печах // "Металлургическая и горнорудная промышленность №1, с.42-47. Днепропетровск, 2009 г.

65. Руткас А.А., Хаханов В.И. Структурные и функциональные характеристики дескрипторных нейронных сетей // Радиоэлектроника и информатика. 2009. №1. С. 37-46.

66. Какурин Н.Я., Лопухин Ю.В., Макаренко А.Н., Замалеев Ю.С. Системное проектирование преобразователей кодов дробных чисел по методу накопления эквивалентов // АСУ и приборы автоматики. 2009. Вып. 146. С. 33-39.

67. Кулак Э.Н., Каминская М.А., Константинова Ю.К. Метод анализа тестопригодности цифровых устройств для псевдослучайного тестирования в системах встроенного самотестирования // АСУ и приборы автоматики. 2009. Вып. 147. С. 9-15.

68. Хаханов В.И., Чумаченко С.В., Литвинова Е.И., Захарченко О.В. Технология покрытия дефектных блоков резервными компонентами // АСУ и приборы автоматики. 2009. Вып. 147. С. 52-64.

69. Хаханов В.И. Элитное образование – модель высшей школы будущего // Научно-методический специализированный журнал "Нова педагогічна думка", 2009. 9 с.

70. Руткас А.А., Хаханов В.И. Структурные и функциональные характеристики дескрипторных нейронных сетей // Радиоэлектроника и информатика. 2009. №3. С. 37-44.

71. Шкиль А.С., Кривуля Г.Ф., Каук В.И., Гаркуша Е.В. Комп'ютерне тестування як технологія контролю залишкових знань // Вестник Херсонского государственного технического университета. - Херсон: ХГТУ – 2009. – №1 (34). С. 520-528.

72. Кривуля Г.Ф., Каук В.И., Шкиль А.С. Контроль залишкових знань // Вища школа – Київ – 2009. - №5. С. 97-110.

73. Шкиль А.С., Сыревич Е.Е., Кучеренко Д.Е., Фастовец Г.П. Методы поиска ошибок проектирования в HDL-коде // Радиоэлектроника и информатика. – Харьков. – 2009. – №. 3. С. 47–53.

74. Шкиль А.С., Сыревич Е.Е., Кучеренко Д.Е., Альмадхоун С. Метод обратного прослеживания для поиска ошибок проектирования в HDL-коде // Радиоэлектроника. Информатика. Управление.– Запорожье ЗНТУ. – 2009.– №2. С. 86-90.

75. Какурин Н.Я., Коваленко С.Н., Лопухин Ю.В., Макаренко А.Н. Способ повышения быстродействия преобразователей кодов на счетчиках // АСУ и приборы автоматики. 2009. вып.145 С.86-95.

76. Белова Н.В., Лобода В.Г., Умяров Р.Я. О системотехническом синтезе специализированного процессора // Збірник наукових праць, Харків: Харківський університет повітряних сил – 2009. Вип.2(20) С.70-73.

77. Дзюндзюк Б.В., Ларченко Л.В., Марченко Л.И., Анпилогов Е.М. Исследование растекания тока для определения опасных зон // Харьков, Сборник научных трудов «Вестник НТУ(ХПИ)», 2009. №15. С. 35-39.
78. Бабич А.В., Хаханов В.И., Мурад Али А. Исследование процессов передачи данных в реальном режиме времени // АСУ и приборы автоматики, №147. 2009. С. 4-8.
79. Хаханов В.И., Литвинова Е.И., Чумаченко С.В., Побеженко И.А., Ngene Christopher // Тестирование и верификация HDL-моделей компонентов SOC. I // Радиоэлектроника и информатика. 2009. №3. С. 45-52.
80. Кривуля Г.Ф., Семенец В.В., Шкиль А.С. Учет результатов тестирования в итоговой оценке накопительной системы оценивания // Сборник научных трудов «Образование и виртуальность - 2009». Харьков – Ялта , 2009 .С. 253-260.
81. Хаханов В.И., Чумаченко С.В., Tiesoura Yves, Галаган С.С. Встроенное диагностирование цифровых систем // Радіоелектронні і комп'ютерні системи. 2009. №7(41). С. 314-318.
82. Vladimir Hahanov, Eugenia Litvinova, Wajeb Gharibi, Olesya Guz. Algebra-Logical Repair Method for FPGA Logic Blocks // Radioelectronics and informatics. 2009. № 2. P. 49-56.
83. Хаханов В.И., Литвинова Е.И., Чумаченко С.В., Побеженко И.А., Ngene Christopher. Тестирование и Верификация HDL-Моделей Компонентов SOC. II // АСУ и приборы автоматики, №148. 2009. С. 26-37.
84. Бабич А.В., Мурад Али А. Модели обратной связи для протокола RTSP // АСУ и приборы автоматики, №148. 2009. С. 71-74.
85. Хаханов В.И., Хаханова И.В., Литвинова Е.И., Гузь О.А. Верификация HDL-кода на основе механизма ассерций // Журнал «Информационные системы на железнодорожном транспорте». 2009. 27 с.
86. Хаханов В., Чумаченко С., Литвинова Е., Захарченко О. Метод минимального покрытия конфигурируемых логических блоков резервными компо-

нентами // Журнал «Информационные системы на железнодорожном транспорте». 2009. 7 с.

87. Хаханов В.И. Процесс модели анализа ассоциативных структур данных / В.И. Хаханов, О.А. Гузь, Ngene Christopher Umerah, В. Ольховой // АСУ и приборы автоматики. – 2010. – Вып. 152. – С.21-26.

88. Хаханов В.И. Инфраструктура вычислительного процесса на основе векторных логических операций / В.И. Хаханов, С.В. Чумаченко, Tiesoura Yves, В.А. Василенко // АСУ и приборы автоматики. – 2010. – Вып. 153. – С.4-12.

89. Хаханов В.И. Инфраструктура диагностирования вредоносных программ в индивидуальном кибернетическом пространстве / В.И. Хаханов, С.В. Чумаченко, А.С. Мищенко, А.В. Зацарный, Ю.В. Хаханова // АСУ и приборы автоматики. – 2010. – Вып. 153. – С.19-32.

90. Бабич А.В. Дисперсионный анализ в процессах пассивного мониторинга и диагностирования корпоративных компьютерных сетей / А.В. Бабич, И.В. Емельянов // АСУ и приборы автоматики. – 2010. – Вып. 153. – С.58-64.

91. Хаханов В.И. Метрика алгебры векторной логики для кибернетического пространства / Хаханов В.И., Мищенко А.С., Вареца В.В. // Радиоэлектроника и информатика. – 2010. – №3. – С. 39-42.

92. Хаханов В.И. Логический метод поиска дефектов по таблице неисправностей / Хаханов В.И., Тиекура Ив (Tiesoura Yves) // Радиоэлектроника и информатика. – 2010. – №3. – С. 43-45.

93. Кривуля Г.Ф. Анализ списка соединений в системах логического синтеза / Кривуля Г.Ф., Сыревич Е.Е., Карасев А.Л. // Радиоэлектроника. Информатика. Управление. – Запорожье ЗНТУ. – 2011. – №2. – С.65-67.

94. Какурин Н.Я., Лопухин Ю.В., Вареца В.В., Катасонов В.В., Макаренко А.Н. Программное средство для анализа характеристик преобразователей кодов по методу досчета // АСУ и приборы автоматики. 2010. Вып. 152. С.41-48.

95. Какурин Н.Я., Лопухин Ю.В., Вареца В.В., Катасонов В.В., Макаренко А.Н. Анализ преобразователей двоично-к-ичного кода в двоичный код по методу досчета // АСУ и приборы автоматики. 2010. Вып. 153. С.45-52.

96. Бабич А.В., Емельянов И.В. Дисперсионный анализ в процессах пассивного мониторинга и диагностирования корпоративных компьютерных сетей // АСУ и приборы автоматики. 2010. Вып. 153. С.58-64.

97. Кривуля Г.Ф., Шкиль А.С., Кучеренко Д.Е., Гаркуша Е.В. Нечеткая логика в экспертной оценке ИКТ-компетентностей // Вісник ХНТУ. Херсон, 2011. № 2 (41). С. 13-23.

98. Кривуля Г.Ф., Сыревич Е.Е., Карасев А.Л. Атрибуты списка соединений в системах логического синтеза // Вісник ХНТУ. Херсон, 2011. № 2 (41). С. 166-171.

99. Кривуля Г.Ф., Шкиль А.С., Кучеренко Д.Е., Гаркуша Е.В. Процедуры диагностирования компетентности пользователей компьютерных систем // Інформаційно - керуючі системи на залізничному транспорті. 2011. №.2. С.17-21.

100. Ngene Christopher Umerah, Литвинова Е.И., Хаханов В.И., Tiesoura Yves. Модели генерации тестов и методы диагностирования SoC-компонентов // Радиоэлектроника и информатика. 2011. №1. С. 64-74.

101. Ngene Christopher Umerah, Хаханов В.И., Зайченко С.А., Литвинова Е.И., Скворцова О.Б. Модели и методы верификации и диагностирования SoC HDL-кода // АСУ и приборы автоматики. 2011. Вып. 154. С.11-32.

102. Какурин Н.Я., Бочаров Е.В., Вареца В.В., Полежаев К.В., Замалеев Ю.С. Программное средство для анализа преобразований чисел в преобразователях кодов параллельного типа // АСУ и приборы автоматики. 2011. Вып. 154. С.83-90.

103. Какурин Н.Я., Лопухин Ю.В. Проектирование преобразователей кодов по методу досчета на VHDL // Інформаційно - керуючі системи на залізничному транспорті. 2011. №. 4. С.64-67.

104. Кривуля Г.Ф., Сыревич Е.Е., Карасев А. Изменение уровня иерархии запросов при проектировании цифровых систем на кристаллах // Інформаційно-керуючі системи на залізничному транспорті. 2011. №5. С. 22-25.

105. Хаханов В.И., Чумаченко С.В., Литвинова Е.И., Мищенко А.С., Адамов А.С. Инфраструктура анализа и информационной безопасности киберпространства // Радиоэлектроника и информатика. 2011. №2. С. 40-60.

106. Adamov.A., Hahanov V., Wajeb Gharibi. Security risks and modern cyber security technologies for corporate networks // Radioelektroniks and informatics. 2010. № 4. P. 31-35.

107. Hahanov V., Wajeb Gharibi, Litvinova E., Chumachenko S. Models and methods for verification and diagnosis of SoC HDL-code // Radioelektroniks and informatics. 2010. № 4. P. 36-46.

108. Obrizan V. A method of high-level synthesis and verification with system C language // Radioelektroniks and informatiks. 2010. № 4. P. 47-50.

109. Hahanov V., Chumachenko S., Mostovaya K. Metrics of vector logic algebra for cyber space // Radioelektroniks and informatics. 2011. № 4. P. 11-14.

110. Hahanov V., Chumachenko S., Mischenko A. Testing and diagnosis of bad messages in individual cyber space // Radioelektroniks and informatics. 2011. № 4. P. 64-71.

111. Barannik V.V., Hahanova A.V. The method of binary structures compression on basis of cascade encoding in telecommunication systems // Radioelektroniks and informatics. 2011. № 4. P. 96-99.

112. Hahanov V., Wajeb Gharibi, Litvinova E., Chumachenko S. Information analysis infrastructure for diagnosis // Information an international interdisciplinary journal. 2011. Japan. Vol.14. № 7. P. 2419-2433.

113. Hahanov V. East-West Design and Test Symposium // Design and test of Computers.2011. USA. No 6. С. 134.

114. Хаханов В.И., Хаханова И.В., Литвинова Е.И., Гузь О.А. Тестирование и верификация HDL-моделей цифровых систем на кристаллах // Электронное моделирование. 2010. № 2(32). С. 31-39.

Підготовлено та випущено 3 тематичні номери науково-технічного журналу “Біоніка інтелекту” (1(72), 2(73), 3(74)).

4.4. Тези доповідей на конференціях

1. Leshchinskaya I.A., Chetverikov G.G., Vechirskaya I.D. Methods of Synthesizing Reversible Spatial Multivalued Structures of Language Systems // International Book Series “Information Science and Computing”. – 2009. – Vol. 3 – P.32-39.
2. Bondarenko M.F., Chetverikov G. G., Vechirskaya I.D. Multiple-valued structures of Intellectual Systems // International Book Series “Information Science and Computing”. – 2008. – Vol. 2, Number 5 – P.125-130.
3. Вечирская И.Д., Четвериков Г.Г., Федорова Т.Н., Токарев В.В., Лещинская И.О. Концептуально-методологічний підхід до моделювання природної мови алгебро-логічними засобами // Доклады Международной научной конференции “Горизонты прикладной лингвистики и лингвистических технологий” (MegaLing’2009). – 20-27 сентября 2009, Украина, Киев – С. 68.
4. Четвериков Г.Г., Федорова Т.Н. Побудова ієрархії дескрипторів на прикладі іменників української мови // Искусственный интеллект. Интеллектуальные системы: Материалы X Международной научной молодежной школы «ССИИ». – Донецк: ИПИИ «Наука і освіта». – 2009. – Т. 1. – С. 64-65.
5. Четвериков Г.Г., Федорова Т.Н. Побудова гіперланцюгів іменників української мови за допомогою інтегрованої лексикографічної системи // Тези доповідей другої факультетської науково-практичної школи-семінару студентів, аспірантів і молодих науковців “Інформаційні інтелектуальні системи, 8-9 грудня: ХНУРЕ, 2009. -2009” – С. 210-213.
6. Четвериков Г.Г., Згривець М.В. Еліптичні алгоритми як засіб верифікації недоторканності даних при дистанційній передачі// Тези доповідей другої факультетської науково-практичної школи-семінару студентів, аспірантів і молодих науковців “Інформаційні інтелектуальні системи, 8-9 грудня: ХНУРЕ, 2009. -2009” – С. 172-174.
7. Вечірська І.Д. Застосування алгебро-логічних методів для розширення можливостей технологій електронних мультимедійних видань// Матеріали конференції “Проблеми й перспективи розвитку ІТ-індустрії”. – 18-19 листоп. 2009, Україна, Харків. – С. 222-224.
8. Русакова Н.Е. Методы реляционного программирования // Матеріали

- конференції “Проблеми й перспективи розвитку ІТ-індустрії”. – 18-19 листоп. 2009, Україна, Харків. – С. 250-252.
9. Шабанов-Кушнарєнко Ю.П., Дрюк А.Д. Аксиоматическая характеристика основных понятий теории действительных чисел // Тези доповідей другої факультетської науково-практичної школи-семінару студентів, аспірантів і молодих науковців “Інформаційні інтелектуальні системи, 8-9 грудня: ХНУРЕ, 2009. -2009” – С. 168-171.
 10. Шостак И.В., Киселева Я.Г. Интеллектуальная система поддержки принятия решений по борьбе с крупными лесными пожарами // Тези доповідей другої факультетської науково-практичної школи-семінару студентів, аспірантів і молодих науковців “Інформаційні інтелектуальні системи, 8-9 грудня: ХНУРЕ, 2009. -2009” – С. 177-178.
 11. Шабанов-Кушнарєнко Ю.П., Кругликова Н.П. О математическом описании субъективных состояний человека как психофизического объекта // Тези доповідей другої факультетської науково-практичної школи-семінару студентів, аспірантів і молодих науковців “Інформаційні інтелектуальні системи, 8-9 грудня: ХНУРЕ, 2009. -2009” – С. 179-181.
 12. Шабанов-Кушнарєнко Ю.П., Лещинская И.А., Русакова Н.Е. Аппаратная реализация модели реляционной сети памятью с ассоциативным доступом // Тези доповідей другої факультетської науково-практичної школи-семінару студентів, аспірантів і молодих науковців “Інформаційні інтелектуальні системи, 8-9 грудня: ХНУРЕ, 2009. -2009” – С. 182-187.
 13. Шабанов-Кушнарєнко С.Ю., Лещинський В.О. Формальні методи моделювання реляційних мереж // Тези доповідей другої факультетської науково-практичної школи-семінару студентів, аспірантів і молодих науковців “Інформаційні інтелектуальні системи, 8-9 грудня: ХНУРЕ, 2009. -2009” – С. 188-189.
 14. М.Ф. Бондаренко, Н.Е. Русакова. Построение математической модели склонения устных, потенциальных, полных непряжательных имен прилагательных // Тези доповідей 17-ї міжнародної конференції з автоматичного управління „Автоматика-2010”. Том 2. – Харків: ХНУРЕ, 2010. – С.92-93.
 15. М.Ф. Бондаренко. Харьковский национальный университет радиоелектроники-научно-образовательный центр в области радиоелектроники и информатики в Украине // Тези доповідей 17-ї міжнародної конференції з автоматичного управління „Автоматика-2010”. Том 1. – Харків: ХНУРЕ, 2010. – С.12-14.
 16. М.Ф. Бондаренко, Н.Е. Русакова, Шабанов-Кушнарєнко Ю.П. О мозгоподобных структурах // Тези доповідей 17-ї міжнародної конференції з автоматичного управління „Автоматика-2010”. Том 2. – Харків: ХНУРЕ, 2010. – С.94-95.
 17. Вечірська І.Д., Дудар З.В., Токарев В.В., Четвериков Г.Г. Розв’язання задачі знаходження гіпотетично зв’язаних об’єктів // Тези доповідей

- 17-ї міжнародної конференції з автоматичного управління „Автоматика-2010”. Том 2. – Харків: ХНУРЕ, 2010. – С.96-97.
18. Лещинская И.А. Словосочетания естественного языка как реляционные сети // Тези доповідей 17-ї міжнародної конференції з автоматичного управління „Автоматика-2010”. Том 2. – Харків: ХНУРЕ, 2010. – С.123-125.
19. I.D. Vechirska, G.G Chetverikov. Method of solving the problem of finding hypothetically connected objects // Сборник научных трудов IV Международной научной конференции «Функциональная база нанoeлектроники». – 2011. – С. 271-274.
20. М.Ф. Бондаренко, Н.Е. Русакова, Ю.П. Шабанов-Кушнарeнко. Методы логического анализа и синтеза мозгоподобных структур // 18-та Міжнародна конференція з автоматичного управління "Автоматика – 2011" : тези доповідей. – Львів, 2011. – С. 341.
21. Русакова Н.Е. О предикате дифункциональности // Материалы 15-го юбилейного Международного молодежного форума "Радиоэлектроника и молодежь в XXI веке" : зб. матеріалів форуму. – Харків: ХНУРЕ, 2011. – С. 27–28.
22. "Автоматизированная система измерения больших токов". Сліпченко М.І., Єлаков С.Г., Руденко О.Г. Сотников О.М. "Каталог форума". ХНУРЭ, 8-11 октября 2009г. С. 90.
23. "Система контроля температурного режима в трубчатой вращающейся печи". "Каталог форума". ХНУРЭ, 8-11 октября 2009г. Сліпченко М.І., Єлаков С.Г., Руденко О.Г. Сотников О.М. С. 88.
24. Єлаков С.Г., Руденко О.Г., Сотников О.М. Система предварительной обработки, ввода в ПК и анализа изображений нанообъектов от голографического микроскопа IV Международная научная конференция «ФУНКЦИОНАЛЬНАЯ БАЗА НАНОЭЛЕКТРОНИКИ». 30 сентября - 3 октября 2011г. Кацевели Харьков.
25. "Комплекс "Зонд" автоматизированного контроля эффективности текущих электроплавов в руднотермических электропечах". "Каталог форума". ХНУРЭ, 8-11 октября 2009г. Сліпченко М.І., Єлаков С.Г., Руденко О.Г. Сотников О.М. С. 83.
26. Hahanov V. Embedded Diagnosis and Repairing of SOC Memory / V. Hahanov, E. Litvinova, Ngene Christopher Umerah, O. Guz // Proceedings of the Xth International Conference CADSM 2009. – Lviv-Polyana. – 24-28 February 2009. – P.296-300.
27. Hahanov V. Embedded Testing for SOC Functionality / V. Hahanov, S. Pokrova, T. Yves, A. Gorobets // Proceedings of the Xth International Conference CADSM 2009. – Lviv-Polyana. – 24-28 February 2009. – 4 p.
28. Hahanov V. Vector-logical approach to diagnosis of SOC components / V. Hahanov, V. Vasilenko, N. Kulbakova, W. Gharibi // Proceedings of the Xth International Conference CADSM 2009. – Lviv-Polyana. – 24-28 February 2009. – 4 p.

29. Chumachenko S. Graph-based method for Software components testing / S. Chumachenko, S. Galagan, K. Krasnoyaruszhskaya, A. Sushanov // Proceedings of the Xth International Conference CADSM 2009. – Lviv-Polyana. – 24-28 February 2009. – 5 p.
30. Hahanov V. Descriptor neural networks with arbitrary characteristic index / V. Hahanov, A. Rutkas // Proceedings of the Xth International Conference CADSM 2009. – Lviv-Polyana. – 24-28 February 2009. – 4 p.
31. Sushanov A. SoC Functional Intellectual Property Diagnosis / A. Sushanov, V. Varetza, S. Dementyev, V. Vasilenko // Proceedings of the Xth International Conference CADSM 2009, Lviv-Polyana, 24-28 February 2009, 5 p.
32. Melnikova O. Using Multi-FPGA Systems for ASIC Prototyping / O. Melnikova, K. Mostovaya // Proceedings of the Xth International Conference CADSM 2009. – Lviv-Polyana. – 24-28 February 2009. – 5 p.
33. Filippenko Igor Olegovich. Digital communication system // Proceedings of the Xth International Conference CADSM 2009, Lviv-Polyana, 24-28 February 2009, 5 стр.
34. Чумаченко С.В. Тестирование неисправностей цифровых систем на кристаллах с помощью метода дихотомии / С.В. Чумаченко, А.В. Сушанов, С.С. Галаган, О.В. Щерба // Материалы международной научной конференции «Интеллектуальные системы принятия решений и проблем вычислительного интеллекта» (ISDMСГ09). – Евпатория. – 18-22 мая 2009. – Т.1. – С. 112-122.
35. Кривуля Г.Ф. Диагностика компьютерных неисправностей с использованием нечетких экспертных систем / Г.Ф. Кривуля, З.В. Дударь, Д.Е. Кучеренко, М.А. Лаптев // Материалы международной научной конференции «Интеллектуальные системы принятия решений и проблем вычислительного интеллекта» (ISDMСГ09). – Евпатория. – 18-22 мая 2009. – Т. 1. – С. 187-188.
36. Хаханов В.И. Диагностирование программных и аппаратных компонентов цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь, А.А. Победа // Материалы международной научной конференции «Интеллектуальные системы принятия решений и проблем вычислительного интеллекта» (ISDMСГ09). – Евпатория. – 18-22 мая 2009. – Т.1. – С. 225-230.
37. Хаханов В. Энергосберегающие технологии для программно-аппаратной имплементации систем вычислительного интеллекта / В. Хаханов, Ngene Christopher Umerah, Tiesoura Yvees // Материалы международной научной конференции «Интеллектуальные системы принятия решений и проблем вычислительного интеллекта» (ISDMСГ09). – Евпатория. – 18-22 мая 2009. – Т.2. – С. 439-458.
38. Хаханов В.И. Метод восстановления работоспособности цифровых систем на кристаллах на основе FPGA / В.И. Хаханов, А.В. Сушанов, О.А. Гузь, А.А. Горобец // Международная научно-техническая кон-

- ференция DeSSert'2009. – Кировоград. – 2009. – 4 с.
39. Хаханов В.И., Литвинова Е.И., Ngene Christopher Umerah (Nigeria). Модели и методы тестирования современных цифровых систем на кристаллах // Международная научно-техническая конференция DeSSert'2009. – Кировоград. – 5 с.
 40. Хаханов В.И. Встроенное диагностирование цифровых систем / В.И. Хаханов, Е.И. Литвинова, Tiesoura Yvees, С.С. Галаган // Международная научно-техническая конференция DeSSert'2009. – Кировоград. – 2009. – 6 с.
 41. Кривуля Г.Ф., Кучеренко Д.Е., Механа Сами. Классификационные признаки для диагностики компьютерных неисправностей с использованием нечетких экспертных систем // Международная научно-техническая конференция DeSSert'2009, Кировоград, с.127-131.
 42. Механа Сами. An expert system for computer failure diagnosis // Международная научно-техническая конференция DeSSert'2009, Кировоград, с.150-153.
 43. Чумаченко Светлана, Литвинова Евгения, Щерба Ольга, Tiesoura Yves. Восстановление работоспособности функциональных логических блоков FPGA // XI научно-практическая международная конференция «Информационные технологии в образовании и управлении». – 28 -30 мая, 2009. – Новая Каховка, 5 стр.
 44. Кривуля Г.Ф., Шкиль А.С., Каук В.И., Гаркуша О.В. Компьютерное тестирование как технология контроля остаточных знаний // Вестник Херсонского национального технического университета, Херсон 2009, стр. 520-528.
 45. Кривуля Г.Ф., Дударь З.В., Кучеренко Д.Е., Механа Сами. Fuzzy expert System for diagnosis of computer failures // Научно-техническая конференция CADSM`2008 Поляна-Свалява (Закарпатье), 24-28 февраля 2009, с.225-230.
 46. Кривуля Г.Ф. Компьютерное тестирование в накопительной системе оценивания знаний студентов // Материалы XVIII Международной конференции «Вітчизняний та зарубіжний досвід упровадження Булонської системи і успіхи проблеми», Ужгород-Мішкольц, 5-8 травня, 2009, 2 стр.
 47. Хаханов В.И. Элитное образование – модель высшей школы будущего // Матеріали 9 Міжнародної науково-методичної конференції «Проблеми управління якістю підготовки фахівців в умовах інтеграції в міжнародний освітній простір». Рівне. 2009. С.368-376.
 48. Шкиль А.С., Напрасник С.В., Хоменко Р.В. Портативная версия компьютерной системы тестирования знаний OpenTEST2 // Інформатизація вищих навчальних закладів МВС України. Матеріали. науково-практич. конф., Харків 28 травня 2009 р. – Харків, ХНУВС, 2009. С.20–25.
 49. Шкиль А.С., Гаркуша Е.В. Анализ качества тестовых заданий в ком-

- пьютерной системе тестирования знаний OpenTEST2 // Проблемы функціонування вищих навчальних закладів економічного профілю. Матеріали 1 Всеукр. науково-практич. конф., Полтава 22-23 жовтня 2009 р. – Полтава, ПУСКУ, 2009.С.111-116.
- 50.Шкиль А.С., Кривуля Г.Ф., Семенец В.В. Учет результатов тестирования в итоговой оценке накопительной системы оценивания // Сборник научн. труд. 12-й международ. конф. УАДО Образование и виртуальность–2009. – Харьков-Ялта.– 2009.– Харьков: ХНУРЭ.– 2009. С. 253–261.
- 51.Шкиль А.С., Напрасник С.В. Хоменко Р.В. Виртуальный сервер компьютерной системы тестирования знаний OpenTEST2 // Сборник научн. труд. 12-й международ. конф. УАДО Образование и виртуальность–2009. – Харьков-Ялта.– 2009.– Харьков: ХНУРЭ.– 2009. С. 261–266.
- 52.Шкиль А.С., Егоров А.Б. Малышкина Е.С. Оценка достоверности входного контроля знаний студентов // Сборник научн. труд. 12-й международ. конф. УАДО Образование и виртуальность–2009. – Харьков-Ялта.– 2009.– Харьков: ХНУРЭ.– 2009. С. 222–231.
- 53.Syrevitch Yev., Kucherenko D, Karasyov A., Cheglikov D. Methods of HDL - models diagnosis // Proc. of MEMSTECH, 22-24 April, 2009, Polyana-Svalyava (Zakarpattya), UKRAINE, P. 145-148.
- 54.Ларченко Л.В., Марченко Л.И. Компьютерная модель исследования звукопоглощающих средств защиты // Харьков, НТУ(ХПИ), 9 межвузовская научно-методическая конференция «Безпека людини в сучасних умовах», декабрь, 2 стр.
- 55.Василенко В.А.(рук. Хаханов В.И.) Исследование современных методов верификации цифровых устройств // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.321.
- 56.Шеховцов Д.А. (рук. Хаханова И.В.) Исследование асинхронной иерархической модели управляющего автомата для MQ-кодера // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.322.
- 57.Константинова Ю.К. (рук. Кулак Э.Н.) Исследование тестов пригодности цифровых устройств для псевдослучайного тестирования // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.323.
- 58.Гаркуша Е.В. (рук. Кривуля Г.Ф.) Принципы формирования базы данных системы тестирования знаний OPEN TEST 2 // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.7.
- 59.Коваленко С.Н.(рук. Какурин Н.Я.) Алгоритм построения закона функционирования формирователя эквивалентов (ФЭ) // Материалы 13 Международного молодежного форума «Радиоэлектроника и моло-

- дежь в 21 веке». 2009. часть 2. С.9.
60. Катасонов В.В. (рук. Какурин Н.Я.) Варианты структур формирователей эквивалентов в преобразователях кодов параллельного типа // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.13.
61. Быкова Н.Н. (рук. Какурин Н.Я.) Двухкритериальный алгоритм выбора оптимальных значений величин шагов в преобразователях кодов // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.14.
62. Мостовая К.Л., Денисов Е.А. (рук. Хаханов В.И.) Функциональное тестирование программно-аппаратных комплексов // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.20.
63. Филиппенко И.В. (рук. Хаханова И.В.) Повышение характеристик систем передачи данных в информационных системах // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.23.
64. Горбатюк С.А. (рук. Кривуля Г.Ф., Горбатюк А.Ф.) Алгоритмическая реорганизация обчислювальних моделей // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.23.
65. Бочаров Е.В. (рук. Какурин Н.Я.) Программное средство для проектирования многошаговых преобразователей кодов параллельного типа // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.32.
66. Першин М.А. (рук. Хаханов В.И.) Модели анализа результатов поиска в современных сетевых структурах // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.37.
67. Варца В.В. (рук. Какурин Н.Я.) Анализ структуры формирователя эквивалентов в двух шаговом преобразователе кодов // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.61.
68. Сушанов А.в., Ольховой В. (рук. Хаханов В.И.) Программное средство для восстановления работоспособности FPGA // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.141.
69. Сушанов А.В., Галаган С.С. (рук. Хаханов В.И.) Программное средство для диагностирования SiP // Материалы 13 Международного молодежного форума «Радиоэлектроника и молодежь в 21 веке». 2009. часть 2. С.142.
70. Zaharchenko O. Quasi-Optimal Coverage Method for the Addressable Faulty Cells of Digital Systems-On-Chips by Spares / O. Zaharchenko, S. Chumachenko // 4-а Міжнародна конференція “Сучасні комп’ютерні

- системи та мережі: розробка та використання” – Львів. – 2009. – 3 с.
71. Hahanov V. Algebra-Logical Fault Diagnosis Method for SOC Functional Blocks / V. Hahanov, S. Chumachenko, Wajeb Gharibi, Ngiene Christopher Umerah // Proc. of IFAC Discrete-Event System Design. – Gandia Beach. Spain. – 6-8 October 2009. – P. 44-50.
 72. Hahanov V. Algebra-Logical Diagnosis and Repair Method for SoC Memory / Hahanov V., Eugenia Litvinova, Olesya Guz, Tiecoura Yves. // Proc. of IFAC Discrete-Event System Design. Gandia Beach. Spain. 6-8 October 2009. P. 15-20.
 73. Hahanov V. Technology for Faulty Blocks Coverage by Spares / V. Hahanov, S. Chumachenko, E. Litvinova, O. Zakharchenko, N. Kulbakova // Proc. of IEEE East-West Design and Test Symposium. – Moscow. Russia. – 6-8 October 2009. – P. 353-359.
 74. Hahanov V. Coverage Method for FPGA Fault Logic Blocks by Spares / V. Hahanov, E. Litvinova, Wajeb Gharibi, O. Guz // Proc. of IEEE East-West Design and Test Symposium. – Moscow. Russia. – 18-21 September 2009. – P. 43-47.
 75. Hahanov V. Testing and Verification of HDL-models for SoC components / V. Hahanov, I. Hahanova, Ngiene Christopher Umerah, Tiecoura Yves // Proc. of IEEE East-West Design and Test Symposium. – Moscow. Russia. – 18-21 September 2009. – P. 48-53.
 76. Hahanov V. System in Package. Diagnosis and Embedded Repair / V. Hahanov, A. Sushanov, Y. Stepanova, A. Gorobets // Proc. of IEEE East-West Design and Test Symposium. – Moscow. Russia. – 18-21 September 2009. – P. 348-352.
 77. Babich A.V. The Unicast Feedback Models for Real-Time Control Protocol / A.V. Babich, Murad Ali Abbas // Proc. of IEEE East-West Design and Test Symposium. – Moscow. Russia. – 18-21 September 2009. – P. 360-363.
 78. Hahanov V. Algebra-Logical Repair Method for FPGA Logic Blocks / V. Hahanov, S. Galagan, V. Olchovoy, A. Priymak // Proc. of IEEE East-West Design and Test Symposium. – Moscow. Russia. – 18-21 September 2009. – P. 364-368.
 79. Syrevitch Yevgeniya. The Method of Fault Backtracing for HDL - Model Errors Searching / Y. Syrevitch, A. Karasyov, D. Kucherenko // Proc. of IEEE East-West Design and Test Symposium. – Moscow. Russia. – 18-21 September 2009. – P. 369-372.
 80. Lukashenko O. Handling Control Signals for the Scan Technology / O. Lukashenko, D. Melnik, V. Obrizan. // Proc. of IEEE East-West Design and Test Symposium. – Moscow. Russia. – 18-21 September 2009. – P. 373-376.
 81. Melnik D. Early Detection of Potentially Non-synchronized CDC Paths Using Structural Analysis Technique / D. Melnik, O. Lukashenko, S. Zaychenko // Proc. of IEEE East-West Design and Test Symposium. – Mos-

- cow. Russia. – 18-21 September 2009. – P.411-414.
82. Leonid S. Technology of Cascade Structural Decoding / L. Soroka, V. Barannik, A. Hahanov // Proc. of IEEE East-West Design and Test Symposium. – Moscow. Russia. – 18-21 September 2009. – P. 490-494.
83. Hahanov V. Diagnosis and Repair Technologies for System-in-Package / V. Hahanov, E. Litvinova, Wajeb Gharibi, Olesya Guz // 16th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES-2009). – Польша, Лодзь. – 25-27 June 2009. – 4 с.
84. Hahanov V. Embedded Memory Repair Technique for SOC / V. Hahanov, E. Litvinova, Wajeb Gharibi, O. Guz // Proceedings of 15th International Conference on Information and Software Technologies, IT 2009. – Литва, Каунас. – 23-24 апреля 2009. – С.131-138.
85. Хаханов В.И., Литвинова Е.И., Ngene Christopher Umerah (Nigeria), Победа А.А. Сервисное обслуживание компонентов SIP // Международная конференция «Компьютерные науки и информационные технологии». РФ, Саратов, 1-4 июля 2009, 4 стр.
86. Хаханов В.И., Литвинова Е.И., Щерба О.В., Победа А.А. Технологии сервисного обслуживания компонентов SIP // Международная конференция «Компьютерные науки и информационные технологии». РФ, Саратов, 1-4 июля 2009, 5 стр.
87. Хаханов В.И., Гузь О.А., Tiecoura Yvees (Cote d'Ivoire), Щерба О.В. Метод восстановления работоспособности функциональных логических FPGA // Международная конференция «Компьютерные науки и информационные технологии». РФ, Саратов, 1-4 июля 2009. 6 стр.
88. Hahanov V. Logic vector analysis of associative tables / V. Hahanov, S. Chumachenko, E. Litvinova // Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD), 2010. XIth International Workshop. – Gammarth, Tunisia. – 4-6 Oct. 2010. – P. 1-6 This paper appears in: [Symbolic and Numerical Methods, Modeling and Applications to Circuit Design \(SM2ACD\), 2010 XIth International Workshop on](#) Issue Date: 4-6 Oct. 2010 On page(s): 1 - 6 Location: Gammarth, Tunisia Print ISBN: 978-1-4244-6816-4 Digital Object Identifier: 10.1109/SM2ACD.2010.5672300. Date of Current Version: 17 Декабрь 2010.
89. Gharibi Wajeb. SoC Design Verification Infrastructure / Gharibi Wajeb, V. Hahanov // Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD), 2010. XIth International Workshop. – Gammarth, Tunisia. – 4-6 Oct. 2010. – P. 1-6. This paper appears in: [Symbolic and Numerical Methods, Modeling and Applications to Circuit Design \(SM2ACD\), 2010 XIth International Workshop on](#) Issue Date: 4-6 Oct. 2010 On page(s): 1 – 6. Location: Gammarth, Tunisia Print ISBN: 978-1-4244-6816-4 Digital Object Identifier: 10.1109/SM2ACD.2010.5672359. Date of Current Version: 17 Декабрь 2010.
90. Hahanov V.I., Gharibi W., Litvinova E.I., Umerah N.C. Logic associative

- multiprocessor for information analysis // Electronics Conference (BEC), 2010 12th Biennial Baltic. – 2010. – P. 169-172. This paper appears in: [Electronics Conference \(BEC\), 2010 12th Biennial Baltic](#). Issue Date: 4-6 Oct. 2010. On page(s): 169 – 172. Location: Tallinn ISSN: 1736-3705. Print ISBN: 978-1-4244-7356-4. INSPEC Accession Number: 11649408. Digital Object Identifier: 10.1109/BEC.2010.5630712. Date of Current Version: 11 Ноябрь 2010.
91. Хаханова Ю.В., Степанова Ю.В. (научный руководитель – Хаханов В.И.) Инфраструктура логического анализа ассоциативных таблиц // Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.4.
 92. Горобец А.А. (научный руководитель – Литвинова Е.И.) Регистр уплотнения для формирования векторного критерия качества // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.5.
 93. Бочаров Е.В. (научный руководитель – Какурин Н.Я.) Программное средство для проектирования многошаговых преобразователей кодов дробных чисел параллельного типа // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.7.
 94. Катасонов В.В. (научный руководитель – Какурин Н.Я.) Программная реализация нахождения оптимального разбиения преобразователей кодов по методу досчета // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.8.
 95. Мищенко А., Бражников А. (научный руководитель – Хаханов В.И.) Метод диагностирования дефектов на основе ассоциативных таблиц // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.10.
 96. Филиппенко И.О. (научный руководитель – Хаханов В.И.). Системы передачи цифровых данных с оптическим каналом связи // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.11.
 97. Дементьев С.П. (научный руководитель – Сыревич Е.Е.) Аппаратная реализация процедуры тестирования цифровых устройств // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.12.
 98. Сухановская А.В., Кульбакова Н.И. (научный руководитель – Хаханов В.И.) Структура мультипроцессора для выполнения векторных логических операций // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.13.
 99. Черкашин С.А. (научный руководитель – Хаханова И.В.) Методы реализации алгоритмов обработки изображений на DSP процессорах //

- Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.15.
100. Сушанов А.В., Галаган С.С. (научный руководитель – Хаханов В.И.) Проектирование устройства для диагностирования цифровых систем с помощью алгебрологического метода // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.16.
101. Коротких Д.В. (научный руководитель – Шкиль А.С.) Исследование базы знаний электронных учебных материалов для дистанционного обучения // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.17.
102. Касаткин А.В., Сухомлинов Д.А. (научный руководитель – Обризан В.И.) Высокоуровневый синтез цифровых систем на основе языка C++ // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.18.
103. Сорудейкин К.А. (научный руководитель – Хаханов В.И.) Природа пространства и времени в приложении к программной инженерии // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.19.
104. Кузнецов Е.С., Давыдов М.Д. (научный руководитель – Хаханова И.В.) Метод распознавания и отслеживания объектов в видеопотоке данных // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.20.
105. Гаврюшенко А.Ю. (научный руководитель – Обризан В.И.) Модели высокоуровневого синтеза цифровых систем на основе языка C++ // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.21.
106. Кубиря А.В., Приходченко Р.С. (научный руководитель – Хаханов В.И.) Получение векторного изображения текста с использованием SVG и векторных данных шрифта // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.22.
107. Сушанов А.В., Галаган С.С. (научный руководитель – Хаханов В.И.) Диагностирование SoC на основе технологии LAMP // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.23.
108. Приходченко Р.С., Сухановская А.В. (научный руководитель – Хаханов В.И.) Архитектура масштабируемого парсера XML-подобного языка // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.24.
109. Павелко Д.И., Сирук А.В. (научный руководитель – Сыревич Е.Е.)

- Модели задания списка соединений при логическом синтезе на языках описания аппаратуры // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.25.
110. Шарко Д.Н. (научный руководитель – Обризан В.И.) Программно-аппаратный комплекс управления удаленным терминалом // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.26.
111. Лукашенко О., Приймак А. (научный руководитель – Хаханов В.И.) Архитектура логического мультипроцессора // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.27.
112. Шарпан И.В., Полупан Д.Д. (научный руководитель – Хаханова А.В.) Методика обоснования кодирования тестовой информации в системах цифровой диагностики // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.28.
113. Шарпан И.В., Полупан Д.Д. (научный руководитель – Хаханова А.В.) Метод одномерного плавающего структурного кодирования двоичных данных // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.29.
114. Мельник Д.В. (научный руководитель – Чумаченко С.В.) Логический мультипроцессор для анализа информации // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.30.
115. Захарченко О.В., Щербин Д.О. (научный руководитель – Чумаченко С.В.) Минимизация покрытия дефектных логических блоков в SoC // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.31.
116. Перес Перес В. (научный руководитель – Хаханов В.И.) Интегральная метрика определения принадлежности // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.32.
117. Мостовая К., Раков К. (научный руководитель – Хаханов В.И.) Поиск минимального покрытия с помощью логического анализа векторов // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.33.
118. Василенко В.А., Побеженко И.А. (научный руководитель – Хаханов В.И.) Метод верификации HDL-кода на основе транзакционного логического графа // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.34.
119. Побеженко И.А., Кульбакова Н.И. (научный руководитель – Хаха-

- нов В.И.) Верификация HDL-кода на основе механизма ассерций // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.35.
120. Мищенко А. (научный руководитель – Чумаченко С.В.) Модель логического вычислителя для анализа ассоциативных таблиц // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.36.
121. Василенко В.А., Степанова Ю.В. (научный руководитель – Хаханов В.И.) Диагностирование программных модулей WT SoC // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.37.
122. Горобец А.А., Хаханова Ю.В. (научный руководитель – Литвинова Е.И.) Инфраструктура верификации и тестирования SoC // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.38.
123. Вареца В.В. (научный руководитель – Какурин Н.Я.) Сравнительные оценки различных способов реализации формирователей эквивалентов в преобразователях кодов // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.39.
124. Бурковский М.А., Мова А.Ю. (научный руководитель – Немченко В.П..) Использование зарезервированных полей заголовка ETHERNET-фрейма IEEE 802.11 на этапе шифрования // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.48.
125. Филиппенко И.В. (научный руководитель – Хаханов В.И.) Радиотехнические аспекты построения защищенной информационной системы передачи информации на физическом уровне // Материалы XIV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2010. часть 2. 18-20 марта. С.49.
126. Хаханов В.И., Литвинова Е.И., Ngene Christopher Umerah. Инфраструктура саморазвивающегося компьютерного социума // Материалы XV Международной научно-практической конференции «Актуальные вопросы развития инновационной деятельности». Алушта, 17-21 мая 2010 г. С. 60-68.
127. Хаханов В.И., Чумаченко С.В., Tiesouga Yves, Гузь О.А. Инфраструктура мозгоподобных неарифметических вычислений // Материалы XV Международной научно-практической конференции «Актуальные вопросы развития инновационной деятельности». Алушта, 17-21 мая 2010 г. С. 69-75.
128. Хаханов В.И. Инфраструктура логического ассоциативного мультипроцессора для анализа информационного пространства // Труды научно-технической конференции с международным участием «Компьютерное моделирование в наукоемких технологиях». ХНУ им. В.Н.

- Каразина. Часть 1. С. 332-336.
129. Чумаченко С.В., Булавин Д.А., Пирка О.И., Ступенко А.О. Программная реализация графического интерфейса для метода оптимального покрытия дефектных компонентов // Труды научно-технической конференции с международным участием «Компьютерное моделирование в наукоемких технологиях». ХНУ им. В.Н. Каразина. Часть 1. С. 359-362.
130. Чумаченко С.В., Булавин Д.А., Пирка О.И., Ступенко А.О. Верификация математической модели покрытия дефектных компонентов // Труды научно-технической конференции с международным участием «Компьютерное моделирование в наукоемких технологиях». ХНУ им. В.Н. Каразина. Часть 2. С. 281-284.
131. Шкиль А.С., Напрасник С.В., Хоменко Р.В. Виртуальная установка компьютерной системы тестирования знаний OpenTEST2 // Інформатизація вищих навчальних закладів МВС України. Матеріали. науково-практич. конф., Харків 28 травня 2009 р. – Харків, ХНУВС, 2009. С.12–16.
132. Шкиль А.С., Кривуля Г.Ф., Каук В.І. Опыт применения инновационных технологий обучения в техническом университете // Перспективні напрямки і способи реалізації інноваційних технологій у системі європейської вищої освіти Науковий вісник Закарпатського державного університету. Ужгород:ІВА,ЗакДУ, 2010. С. 284-294.
133. Шкиль А.С., Альмадхоун С. Структурный метод поиска ошибок проектирования в HDL-коде // Збірка матеріалів І науково-тех.конф.молодих вчених. Інформаційні управляючі системи та комп'ютерний моніторинг Донецьк, 19-21 травня 2010. Донецьк. 2010. С.170-173.
134. Хаханов В.И., Литвинова Е.И., Побеженко И.А., Ngene Christopher Umerah. Мультипроцессорная инфраструктура анализа информационного пространства // Матеріали міжнародної наукової конференції «Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту». 17-21 травня 2010. Євпаторія. С. 409-413.
135. Хаханов В.И., Чумаченко С.В., Tiesoura Yves, Мищенко А.С. Алгебра для анализа векторно-логического пространства с помощью мультипроцессорного компьютера // Матеріали міжнародної наукової конференції «Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту». 17-21 травня 2010. Євпаторія. С. 414-418.
136. Хаханов В.И., Хаханова А.В., Tiesoura Yves, Ngene Christopher Umerah. Метрика векторно-логического пространства и ее практическое применение // Материалы конференции "Сучасні інформаційні та інноваційні технології на транспорті MINTT-2010". 25-27 мая. 2010. Херсон. С. 247-250.
137. Хаханов В.И., Чумаченко С.В., Литвинова Е.И., Мищенко А.С. Н-метод минимизации булевых функций // Материалы конференции

- "Сучасні інформаційні та інноваційні технології на транспорті MINTT-2010". 25-27 мая. 2010. Херсон. С. 251-255.
138. Hahanov V., Wajeb Gharibi, Litvinova E., Chumachenko S. Cyber space and brain-like computing // Proc. of IEEE East-West Design and Test Symposium. St. Petersburg. Russia. 17-20 September 2010. P.98-109.
 139. Hahanov V., Wajeb Gharibi, Chumachenko S., Litvinova E. Vector logic analysis of associative matrices // Proc. of IEEE East-West Design and Test Symposium. St. Petersburg. Russia. 17-20 September 2010. P.110-117.
 140. Hahanov V., Litvinova E., Priymak A. Table data structures for cyber space // Proc. of IEEE East-West Design and Test Symposium. St. Petersburg. Russia. 17-20 September 2010. P.118-122.
 141. Hahanov V., Guz O., Ngene Christopher Umerah, Olhovoy V. Process models for analyzing associative data structures // Proc. of IEEE East-West Design and Test Symposium. St. Petersburg. Russia. 17-20 September 2010. P.123-126.
 142. Obrizan V. A method for automatic generation of an RTL-interface from a C++ description // Proc. of IEEE East-West Design and Test Symposium. St. Petersburg. Russia. 17-20 September 2010. P.186-189.
 143. Hahanov V., Mishenko A., Varetsa V. Metrics of vector logic algebra for cyber space // Proc. of IEEE East-West Design and Test Symposium. St. Petersburg. Russia. 17-20 September 2010. P.204-207.
 144. Hahanov V., Hahanova A., Zakaryan V. Cyber space evolution // Proc. of IEEE East-West Design and Test Symposium. St. Petersburg. Russia. 17-20 September 2010. P.208-214.
 145. Hahanov V., Pobizhenko I., Tiecoura Yves. Logical method for detecting faults by fault detection table // Proc. of IEEE East-West Design and Test Symposium. St. Petersburg. Russia. 17-20 September 2010. P.215-217.
 146. Krivoulya G.F., Kucherenko D., Dubinsky N. Smart tools for diagnosis computer failures // Proc. of IEEE East-West Design and Test Symposium. St. Petersburg. Russia. 17-20 September 2010. P.390-394.
 147. Альмадхоун С. Поиск ошибок проектирования в HDL-моделях цифровых устройств // Тези доповідей 17 Міжнародної конференції з автоматичного управління «Автоматика-2010» Т.2.С.88-89.
 148. Городниченко С.А., Кривуля Г.Ф., Шуклин Д.Е. Автоматизация задачи классификации текстовых документов с использованием семантической нейронной сети // Тези доповідей 17 Міжнародної конференції з автоматичного управління «Автоматика-2010» Т.2.С.98-99.
 149. Кривуля Г.Ф., Кучеренко Д.Е. Моделирование компетентности пользователя компьютерной системы // Тези доповідей 17 Міжнародної конференції з автоматичного управління «Автоматика-2010» Т.2.С.116-117.
 150. Хаханов В.И. Инфраструктура мозгоподобных вычислительных процессов // Тези доповідей 17 Міжнародної конференції з автоматич-

- ного управління «Автоматика-2010» Т.2.С.137-139.
151. Хаханов В.И., Литвинова Е.И. Архитектура логического ассоциативного мультипроцессора // Тези доповідей 17 Міжнародної конференції з автоматичного управління «Автоматика-2010» Т.2.С.139-141.
 152. Хаханов В.И., Чумаченко С.В. Инфраструктура векторно-логического анализа // Тези доповідей 17 Міжнародної конференції з автоматичного управління «Автоматика-2010» Т.2.С.142-143.
 153. Давыдов А.А. Интеллектуальная диагностика компьютерных систем // Тези доповідей 17 Міжнародної конференції з автоматичного управління «Автоматика-2010» Т.2.С.99-101.
 154. Альмадхоун С., Шкиль А.С., Сиревич Е.Е. Поиск ошибок проектирования в hdl-моделях цифровых устройств // Міжнародна науково-практична конференція «Інформаційні технології та комп'ютерна інженерія», Вінниця, 19-21 травня 2010 р., 377-378.
 155. Шкиль А.С., Напрасник С.В., Хоменко Р.В. Єдиний тестовий сервер університету на основі OpenTEST2 // Інформатизація вищих навчальних закладів МВС України : Матеріали. науково-практич. конф., Харків 28травня 2009 р. Харків, ХНУВС, 2009. С. 24–28.
 156. Шкиль А.С. Єдиний тестовий сервер ВНЗ як основа якісного проведення тестувань // Інноваційні комп'ютерні технології у вищій школі: Матер. 2-ї наук.-практич. конф., Львів 23-25 листопада 2010 р. – Львів: Вид. Львівської політехніки, 2010. С. 130-135.
 157. Бабич А.В., Емельянов И.В. Исследование корреляционной зависимости между каналом связи и алгоритмами прикладного ПО // Харьков-Киев, материалы 1-й научно-технической конференции «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління». С. 66.
 158. Кривуля Г.Ф., Дубинская Н.Г. Диагностирование компьютерных систем с использованием интеллектуальных средств // 10-я международная н/т конференция «Проблемы информатики и моделирования». Харьков-Ялта.27-29 сентября 2010г. С.4-5.
 159. Давыдов А.А. Оптимизация бинарных деревьев решений в экспертной диагностической системе реального времени // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». Алушта. 2010. С. 40.
 160. Filippenko I.V. Solving the problem of collisions in the RF identification systems // Proc. of the V International Scientific and Technical Conference. Lviv, 2010. P. 60.
 161. Hahanov V., Wajeb Gharibi, Olesya Guz. Brain-like infrastructure for Embedded SoC diagnosis // Proceedings of 2010th International Conference on Automation, Quality and Testing, Robotics.THETA 17th Edition. May 28-30. 2010. Cluj-Napoca. Romania.С. 9-13.
 162. Чумаченко С.В., Захарченко О.В., Галаган С.С. Модель покрытия

- дефектных компонентов SOC // Сборник трудов VI Международной конференции «Стратегия качества в промышленности и образовании» (Варна, Болгария), 4-11 июня 2010 г. С. 450-453.
163. Чумаченко С.В., Мищенко А.С., Галаган С.С. Векторно-матричная алгебра логики для параллельных вычислительных процессов // Сборник трудов VI Международной конференции «Стратегия качества в промышленности и образовании» (Варна, Болгария), 4-11 июня 2010 г. С. 662-665.
164. Хаханов В.И., Литвинова Е.И., Василенко В.А., Сушанов А. Метод неопределенных коэффициентов минимизации булевых функций с использованием сложной процесс-модели // Сборник трудов VI Международной конференции «Стратегия качества в промышленности и образовании» (Варна, Болгария), 4-11 июня 2010 г. С. 654-657.
165. Шкиль А.С., Напрасник С.В., Хоменко Р.В., Гаркуша Е.В. Методика анализ качества тестовых заданий в системе тестирования знаний OPENTEST2 // Сборник трудов VI Международной конференции «Стратегия качества в промышленности и образовании» (Варна, Болгария), 4-11 июня 2010 г. С. 675-678.
166. Hahanov V., Chumachenko S., Litvinova E. Logical analysis of information in tabular form International conference on computational technologies in electrical and electronics engineering "Sibircon - 2010". Irkutsk, Russia. July 11-15. 2010. P. 72-79.
167. Hahanov V., Litvinova E., Pobezhenko I. SoC verification infrastructure // International conference on computational technologies in electrical and electronics engineering "Sibircon - 2010". Irkutsk, Russia. July 11-15. 2010. P. 80-85.
168. Болтонюк К.А., Писаренко Д.В. Научный руководитель – д.т.н., проф. Литвинова Е.И. Автоматизированная обработка проектных данных // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.6-7.
169. Болтонюк К.О., Писаренко Д.В. Научный руководитель – д.т.н., проф. Литвинова Е.И. Автоматизована система розрахунку норм праці // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.8-9.
170. Добарский А. А. Научный руководитель – асс. Обризан В.И. Исследование реконфигурируемого сопроцессора на базе FPGA для ускорения вычислений // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.10-11.
171. Новик С.А. Научный руководитель – д.т.н., проф. Хаханова И.В. Аппаратное ускорение верификации моделей цифровых устройств // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.12-13.

172. Быков М.П. Научный руководитель – асп. Сорудейкин К.А. Анализ кода программной системы с помощью средств пошагового инструментирования // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.19-20.
173. Бурковский М.А. Научный руководитель – к.т.н., доц. Бабич А.В. Использование расширенных заголовков пакетов IPv6 в адаптивной маршрутизации // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.30-31.
174. Кучеренко Д.Е., Зиарманд А.Н. Научный руководитель – к.т.н., доц. Шкиль А.С. Анализ компетентности пользователя компьютерной системы как диагностический эксперимент // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.40-41.
175. Мова А.Ю. Научный руководитель – к.т.н., доц. Бабич А.В. Исследование процедур повышения эффективности передачи данных в мультисервисных сетях // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.42-43.
176. Азаров Г.А., Белоус В.В., Бутенко С.А. Научный руководитель – ст. пр. Зайченко С.А. Верификация на основе ассерций на TLM-уровне моделирования цифровых систем // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.53-54.
177. Бочаров Е.В., Полежаев К.В. Научный руководитель – к.т.н., проф. Какурин Н.Я. Реализация режима преобразования чисел в преобразователях кодов параллельного типа // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.55.
178. Бражников А.Н. Научный руководитель – доц. Лопухин Ю.В. Исследование особенностей задач проектирования распределенных информационных систем в условиях неопределенности исходных данных // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.56-57.
179. Вареца В.В. Научный руководитель – к.т.н., проф. Какурин Н.Я. Математические модели многоблочных формирователей эквивалентов для специализированных преобразователей кодов // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.58-59.
180. Дементьев С.П. Научный руководитель – д.т.н., проф. Кривуля Г.Ф. Модели и методы размещения элементов, полученных на этапе логического синтеза // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20

- апреля. С.66-67.
181. Зенович Е.А. Научный руководитель – к.т.н., доц. Сыревич Е.Е. Модель и метод трассировки многослойных печатных плат мобильных устройств // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.68-69.
 182. Карасев А.Л. Научный руководитель – к.т.н., доц. Сыревич Е.Е. Атрибуты списка соединений в системах логического синтеза при прототипировании // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.70-71.
 183. Комашка А.А., Косилов А.В., Першин С.М. Научный руководитель – ст. пр. Зайченко С.А. Унифицированный метод анализа покрытия проектов цифровых систем // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.72-73.
 184. Коротких Д.В. Научный руководитель – к.т.н., доц. Шкиль А.С. Автоматизация проектирования с использованием фреймовой структуры // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.74-75.
 185. Орехова И.В. Научный руководитель – к.т.н., доц. Сыревич Е.Е. Разработка и оптимизация методов классификации и подбора IP-CORE с использованием правил нечеткой логики // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.78-79.
 186. 19.Альмадхоун Самер. Научный руководитель – к.т.н., доц. Шкиль А.С. Обратное прослеживание при поиске ошибок проектирования в HDL-моделях цифровых устройств // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.80-81.
 187. Ганзен В.В., Кульбакова Н.И. Научный руководитель – д.т.н., проф. Хаханова И.В. Исследование алгоритмов аппаратной реализации быстрого преобразования Фурье // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.93-94.
 188. Сорудейкин К.А. Операционный анализ и степень инерционности в моделировании мыслительной деятельности // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии». 2011. 24-25 мая. С. 335-336.
 189. Хаханов В.И., Зайченко С.А., Литвинова Е.И. Поиск функциональных нарушений в программных блоках // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные тех-

- нологии». 2011. 24-25 мая. С. 391-392.
190. Хаханов В.И., Хаханова А.В., Ngene Christopher Umerah. Матричный метод поиска функциональных нарушений // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии». 2011. 24-25 мая. С. 393-394.
191. Сыревич Е.Е., Карасев А.Л., Альмадхоун С. Изменение уровня иерархии запросов на основании библиотечных элементов систем логического синтеза // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии». 2011. 24-25 мая. С. 403-404.
192. Хаханова И.В., Кулак Э.Н., Савутин А.Ю., Бородин И.Б. Спецпроцессор для обработки логических отношений // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии». 2011. 24-25 мая. С. 409-410.
193. Коротких Д.В. Функциональная структура дистанционного образования с применением ИИ // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии». 2011. 24-25 мая. С. 411-412.
194. Babich A., Murad Ali A. Investigation of correlation between state of link channel and algorithms of applied software // Матеріали XI Міжнародної науково-технічної конференції CADSM 2011. 23-25 лютого. 2011. Львів. С. 157-159.
195. Krivoulya G., Shkil A., Kucherenko D. Preparation of qualifying tasks for diagnosis experiment of computer system's users competence analysis // First international workshop Criss-Dessert 2011. Kirovograd. May 11-13. С. 291-299.
196. Anna Babich, Murad Ali A. Investigation of Correlation between State of Link Channel and Algorithms of Applied Software // Мат-лы 11й междунар. научно-техн. конф. Львов-Поляна. «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці». С. 157-159.
197. Бабич А.В., Емельянов И.В. Модель компьютерной сети как объекта мониторинга // Мат-лы междунар. научно-техн. конференции SAIT 2011. Киев. С. 416.
198. Бабич А.В., Емельянов И.В. Процессы активного мониторинга корпоративных компьютерных сетей // Мат-лы 4-й всеукраинской научно-техн. конф. «Информационные процессы и технологии». 2011. Севастополь. С. 74-76.
199. Melnikova O. Overview of the prototyping technologies for Actel® RTAX-S FPGAs // Proc. of IEEE East-West Design and Test Symposium.

- Sevastopol. 19-20 September 2011. P.90-93.
200. Babich A., Murad Ali A. Dispersion analysis in processes of passive monitoring and diagnosing of enterprise area networks // Proc. of IEEE East-West Design and Test Symposium. Sevastopol. 19-20 September 2011. P.295-298.
 201. Ngene Christopher Umerah, Hahanov V. A diagnostic model for detecting functional violation in HDL-code of SoC // Proc. of IEEE East-West Design and Test Symposium. Sevastopol. 19-20 September 2011. P.299-302.
 202. Krivoulya G., Shkil A., Kucherenko D. Competence as a support factor of the computer system operation // Proc. of IEEE East-West Design and Test Symposium. Sevastopol. 19-20 September 2011. P.303-310.
 203. Sorudeykin K. A model of spatial thinking for computational intelligence // Proc. of IEEE East-West Design and Test Symposium. Sevastopol. 19-20 September 2011. P.311-318.
 204. Хаханов В. И., Зайченко С. А., Чумаченко С. В., Литвинова Е. И., Гузь О. А. Диагностирование модулей программно-аппаратных систем // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С. 47-55.
 205. Какурин Н.Я., Лопухин Ю.В. Проектирование преобразователей кодов по методу досчета на VHDL языке // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С. 64-67.
 206. Кривуля Г.Ф., Шкиль А.С., Гаркуша Е.В. Готовность компьютеризованных систем управления и компетентность пользователя // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С. 12-17.
 207. Немченко В.П., Зиарманд А.Н., Чепелев Ю.А. Моделирование сетевых протоколов при построении тестовых последовательностей // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С. 18-21.
 208. Кривуля Г.Ф., Сыревич Е.Е., Карасев А.Л. Изменение уровня иерархии запросов при проектировании цифровых систем на кристаллах // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С.22-25.
 209. Альмадхоун Самер. Поиск ошибок проектирования в HDL-коде цифровых устройств // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С.53-55.

210. Вареца В.В., Катасонов В.В. Оптимизация быстродействия преобразователей кодов по методу досчета на основе принципа многоблочности // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С.56-59.
211. Гаркуша Е.В. Оценка компетентности пользователей как фактор повышения работоспособности компьютерных систем // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С.60-62.
212. Кривуля Г.Ф. , Сыревич Е.Е. Использование предыдущих проектных решений при автоматизированном проектировании цифровых систем // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С.63-66.
213. Кулак Э.Н., Ларченко Л.В., Хаханова И.В. Специализированный вычислитель с числоимпульсным входным сигналом // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С.67-68.
214. Кулак Э.Н., Ларченко Л.В., Хаханова И.В. Спецвычислитель для логических отношений // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С.69-71.
215. Макаренко А.Н., Лопухин Ю.В. Проектирование преобразователей кодов по методу накопления эквивалентов на VHDL языке // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С.8-10.
216. Хаханова И.В., Кулак Э.Н., Ларченко Л.В. Использование показателей тестопригодности для генерации взвешенного теста // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С.11-13.
217. Немченко В. П., Изотов А. С. Построения системы генерации тестовых последовательностей для сетевых протоколов // Конференция «Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины». 23-29 сентября. 2011. Алушта. С. 73-80.
218. Сорудейкин К.А. Субстациональные структуры в процессорных элементах // 4-й Международный радиоэлектронный форум «Прикладная радиоэлектроника. Состояние и перспективы развития». Харьков.18-21 октября 2011. С. 126-130.
219. Бабич А.В., Емельянов И.В. Дисперсионный анализ в процессах

пассивного мониторинга и диагностирования корпоративных компьютерных сетей // Мат-ли 2-ї Міжнар. науково-техн. конф-ії «Інформаційні технології в навігації і управлінні: стан та перспективи розвитку». Київ. 2011. С. 37.

220. Кривуля Г.Ф., Сыревич Е.Е., Карасев А. Использование предыдущих проектных решений при автоматизированном проектировании цифровых систем // Материалы докладов 24-й международной конференции “Перспективные компьютерные, управляющие и телекоммуникационные системы для железнодорожного транспорта Украины”. С 100-101.
221. Хаханов В.И., Чумаченко С.В., Ив Тиекура, Василенко В.А. Логический ассоциативный анализ информации // Материалы 7-й Международной конференции «Автоматизация проектирования дискретных систем». 16-17 ноября 2010. Минск. С. 23-33.
222. Хаханов В.И., Литвинова Е.И., Умерах Нгне Кристофер, Побеженко И.А. Архитектура логического ассоциативного мультипроцессора // Материалы 7-й Международной конференции «Автоматизация проектирования дискретных систем». 16-17 ноября 2010. Минск. С. 34-44.

4.5. Договори з зацікавленими організаціями

Укладено наступні госпдоговори та договори про науково-технічне співробітництво:

- 1) Госпдоговір із ЗАТ «Лабораторія Касперських» (Росія, Москва) про надання інформаційно-консультаційних послуг №09-29 від 05.05.2009.
- 2) Договір з Донецькою академією залізничного транспорту про надання науково-технічних послуг №09-21 від 27.04.2009.
- 3) Договір про науково-технічне співробітництво між ХНУРЕ та фірмою ALDEC Inc. (США) від 09.12.2009.
- 4) Госпдоговір №09-55 з німецькою фірмою на 10 тис. євро (Замовник IGUS – ITS Gmb в особі керуючого П. Фолькмера, діє з 1 грудня 2009р. по 31 серпня 2010 р.).
- 5) Договір з Донецькою академією залізничного транспорту про надання науково-технічних послуг № 05/09 від 20.05.2010.
- 6) Договір про співробітництво з Університетом Бумердес (Алжир) – грудень 2010р.

7) Договір про співробітництво з Державним університетом Вірменії (Єреван) від 21.12.2010.

8) Укладено г/д НДР № 10-37 «Розроблення лексикографічної системи Українсько-російсько-англійського словника з електроніки та інформатики».

9) Договір про співробітництво між ХНУРЕ та Плімутським університетом (Велика Британія) від 25.12.2010.

4.6. Організація та проведення конференцій та семінарів

З 18 по 21 вересня 2009 р. у м. Москва, Росія, за ініціативою ХНУРЕ та за підтримки Московського інституту електроніки та математики, IEEE Computer Society Test Technology Technical Council – ТТТС, а також за фінансової підтримки компаній Cadence, JTAG Technologies, Kaspersky Lab, Synopsys, Mentor Graphics, Tallinn University of Technology, Donetsk Institute of Road Transport, Virage Logic, Echostar, Tetrocomp, DataArt Lab, відбувся 7-й Міжнародний науковий симпозиум «East-West Design & Test Symposium - 2009». Рейтинг симпозиуму – № 6 серед прем'єр-конференцій планети з комп'ютерної інженерії. До участі у симпозиумі було прийнято 134 доповіді від 375 авторів з 36 країн світу. Кількість учасників – 131. Кількість пленарних та замовлених доповідей – 18. Кількість університетів та компаній – 88. Кількість міст – 69. Були видані програма та матеріали конференції (Proceedings) обсягом 546 стор. формату А4. За результатами проведення симпозиуму вийшло 3 програми «Дзеркало науки» від каналу ОТБ (Харківське обласне телебачення).

З 17 до 21 вересня 2010 р. у м. Санкт-Петербург, Росія, за ініціативою Харківського національного університету радіоелектроніки та за підтримки Ленінградського електротехнічного інституту (ЛЕТІ), IEEE Computer Society Test Technology Technical Council – ТТТС відбувся 8-й Міжнародний науковий симпозиум «East-West Design & Test Symposium - 2010». Рейтинг симпозиуму – № 6 серед прем'єр-конференцій планети з комп'ютерної інженерії. До участі у симпозиумі було прийнято 111 докладов от 216 авторов из 34 стран мира (Armenia, Azerbaijan, Bangladesh, Belarus, Brazil, Côte d'Ivoire, Croatia, Egypt, Estonia,

Finland, France, Germany, India, Iran, Israel, Italy, Japan, Jordan, Luxembourg, Mexico, Nigeria, Norway, Poland, Colombia, Canada, Romania, Russia, Saudi Arabia, Singapore, Syria, Sweden, UK, Ukraine, USA). Количество участников - 101. Количество пленарных и заказных докладов - 16. Були видані програма та матеріали конференції (Proceedings) обсягом 546 стор. формату А4. За результатами проведення симпозиуму вийшло 3 програми «Дзеркало науки» від каналу ОТБ (Обласне телебачення).

У 2010 р. також проведено наступні наукові семінари та конференції:

1. «Мозкоподібні структури» (наук. кер.: д.т.н, проф. Бондаренко М.Ф., д.т.н, проф. Шабанов-Кушнарченко Ю.П.)

2. 17-та міжнародної конференції з автоматичного управління-2010» та видання тез доповідей (відповідальні випускові: д.т.н., проф. Четвериков Г.Г., к.т.н. Вечірська І.Д.).

9-12 вересня 2011 р. проведено IEEE EWDTs-2011 у Севастополі (Україна) за участю студентів та аспірантів (за підтримки компаній-спонсорів Aldec, Synopsys, Cadence, Kaspersky Laboratory, DataArt Lab, Tallinn Technical University). До участі у симпозиумі було прийнято 101 доповідь від 222 авторів из 34 країн світу (Armenia, Belarus, Brazil, Czech Republic, Côte d'Ivoire, Egypt, Estonia, Finland, Germany, India, Iran, Israel, Italy, Japan, Korea, Lebanon, Libyan Arab Jamahiriya, Malaysia, Mexico, Nigeria, Norway, Palestine, Poland, Romania, Russia, Saudi Arabia, Singapore, Syria, Sweden, Turkey, UK, Ukraine, USA, Viet Nam). Кількість учасників – 101. Кількість пленарних та замовлених доповідей – 12. Кількість університетів та компаній – 82. Видано програму та збірку праць симпозиуму (Proceedings) обсягом 420 стор. формату А4 із включенням до IEEE-Library доповідей, що були представлені на симпозиумі, серед них – 11 доповідей з ХНУРЕ; за результатами проведення симпозиуму підготовлено 3 програми «Дзеркало науки» та «Європоступь» від Харківського обласного телебачення (ОТБ), що популяризують досягнення вчених ХНУРЕ.

4.7. Отримані нагороди

Д.т.н., проф. Хаханов В.І. був нагороджений Медаллю «За наукові досягнення» (наказ № 921-к від 28.09.2009, посвідчення № 1199).

4.8. Отримані гранти

У 2009 році отримано 70 грантів:

- на проведення наукових досліджень – 1 (Лабораторія Касперських);
- на участь у конференції/семінарі/круглому столі за кордоном – 39 (від компаній-спонсорів JTAG Technologies, Kaspersky Lab, Cadence, Synopsys, Mentor Graphics, Tallinn Technical University, Virage Logic, Echostar, Aldec, Terprocomp, DataArt Lab, Intel, Microsoft);
 - на стажування/підвищення кваліфікації – 9 (EchoStar, Плімутський університет, City Media AS / Starlife (Норвегія);
 - на навчання – 1 (Посольство Кореї);
 - на організацію науково-навчальних заходів – 20 (від компаній-спонсорів Echostar, Aldec, Kaspersky lab, DataArt lab, Tallinn TU на проведення Міжнародної олімпіади з технічної діагностики обчислювальних пристроїв і систем).

У 2010 році отримано 33 гранти:

1) на участь у конференції/семінарі/круглому столі – 13:

1 грант від центру розробок компанії DataArt – на участь у 10-й зустрічі ІТ-співтовариства IT Talk 28.01.2010 р.; 1 грант на участь у конференції «Підвищення якості освіти» 15 – 20 травня 2010 р. (Словачія) – д-р техн. наук, проф. Кривуля Г.Ф.; 2 гранти на участь у Міжнародній науковій конференції «IT Security for the Next Generation» в м. Краків, Польща, від об'єднаного оргкомітету - Лабораторія Касперського (Росія, Москва) та Ягелонського університету (Jagiellonian University, м. Краків, Польща) – д.т.н., проф. Хаханов В.І., д.т.н., проф. Чумаченко С.В.; 2 гранти на участь у Міжнародній науковій конференції IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics AQTR в м. Клуж-Напока, Румунія, 26-30 травня 2010 р. – д.т.н., проф. Хаханов В.І., д.т.н., проф. Чумаченко С.В.; 1 грант на участь у семінарі «Дні

Лабораторії Касперських у Києві», квітень 2010 р. – д-р техн. наук, проф. Хаханов В.І.; 1 грант на участь у конференції «СИБИРКОН» від оргкомітету 10-15 липня 2010 р. – д-р техн. наук, проф. Хаханов В.І.; 5 грантів на участь у EWDTTS-2010 у Санкт-Петербурзі (Росія) 18-21 вересня 2010 р. за участю студентів та аспірантів: від компаній-спонсорів Virage Logic, Aldec, Synopsys, Kaspersky Laboratory, DataArt Lab, Tallinn Technical University, JTAG Technologies и Донецької академії автомобільно-дорожнього транспорту.

2) на стажування/підвищення кваліфікації – 5: 2 гранти на стажування в Плімутському університеті (Великобританія, грудень 2009 р. – січень 2010 р., грудень 2010 р. – січень 2011 р.) – д-р техн. наук, проф. Кривуля Г.Ф.; 1 грант на стажування у компанії City Media AS / Starlife (Норвегія): 17-23.01.2010 Валерій Шипунов, студент гр. СКС-05-1; 2 гранти на стажування в компанії Aldec (США), м. Лас-Вегас у рамках договору про співробітництво від 09.12.2009 аспіранту стаціонара 2-го року навчання Мельнику Д.В. Терміни: 12.02.2010–12.04.2010, 02.10.2010–02.12.2010.

3) на організацію науково-навчальних заходів – 15: 6 грантів від компаній-спонсорів Echostar, Aldec, Kaspersky Lab, DataArt Lab, DataArt, Tallinn TU на проведення Міжнародної олімпіади з технічної діагностики обчислювальних пристроїв і систем 18-20 березня 2010 р. (кількість країн-учасниць – 3 (Україна, Росія, Вірменія); кількість університетів-учасників – 18; число студентів-учасників – 53; число викладачів і кураторів – 14). Всі 90 учасників відзначені грамотами, дипломами та цінними подарунками; 1 грант на участь в олімпіаді з мікроелектроніки в м. Єреван (Вірменія) 25.10.2010-01.11.2010 – студент гр. КІ-07-8 Кубиря А.; 8 грантів на організацію та проведення EWDTTS-2010 у Санкт-Петербурзі (Росія) 18-21 вересня 2010 р. за участю студентів та аспірантів: від компаній-спонсорів Virage Logic, Aldec, Synopsys, Kaspersky Laboratory, DataArt Lab, Tallinn Technical University, JTAG Technologies и Донецької академії автомобільно-дорожнього транспорту.

У 2011 році отримано 4 гранти:

1 грант на участь у симпозиумі «IEEE VLSI Test Symposium» від оргкомітету 1-5 травня 2011 р. – д-р техн. наук, проф. Литвинова Є.І. Мета – участь з доповіддю «Infrastructures for verification and embedded diagnosis of digital systems-on-chips»;

3 гранти на організацію та проведення IEEE EWDTs-2011 у Севастополі (Україна) 9-12 вересня 2011 р. за участю студентів та аспірантів: від компаній-спонсорів Aldec, Synopsys, Cadence.

4.9. Дисертації та впровадження результатів

Захищено 12 кандидатських дисертацій та 1 докторська:

1. Іванілов А.О. Реляційні засоби побудови моделей логічних мереж та їх застосування в штучному інтелекті, 2009 (к.т.н.)

2. Ситнік Л.Г. Методи логічного синтезу багатозначних оборотних неоднорідних модулів та їх застосування в системах штучного інтелекту, 2010 (к.т.н.)

3. Супрун Т.С. Інтегральні моделі компараторної ідентифікації та їх застосування для розпізнавання зорової інформації, 2010 (к.т.н.)

4. Процай Н.Т. Кванторна алгебра та її застосування в штучному інтелекті, 2011 (к.т.н.)

5. Побіженко І.О., науковий керівник д.т.н., проф. каф. АПОТ Хаханов В.І. Впровадження результатів – арифметичний блок з раціональними та перегорнутими раціональними коефіцієнтами системи DSP SoC, що реалізує стандарт стиснення зображень JPEG2000: у навчальний процес ХНУРЕ (акт впровадження від 28.19.2009); у науково-дослідну та виробничу діяльність ВАТ «АТ НДРВ» (довідка № 9 від 15.07.2009); виробничий процес фірми «Aldec, Inc» (довідка від 26.09.2009).

6. Зайченко С.О. «Верифікація цифрових систем на кристалах з використанням механізмів асерцій», науковий керівник д.т.н., проф. каф. АПОТ Хаханов В.І. Впровадження результатів: програмний комплекс AKERNEL, що реалізує метод функціональної верифікації систем на кристалах з аналізом темпора-

льних асерцій у науково-дослідницьку та виробничу діяльність ВАТ «АО НИИРИ» (довідка про впровадження); аналітична модель верифікації та інфраструктури процесу верифікації та діагностування у навчальний процес ХНУРЕ (акт впровадження від 12.07.2010 р.); сертифікат компанії Aldec (USA) щодо технологічного вкладу Зайченко С.О. у прикладні розробки асерційного проекту як провідного розробника та автора інноваційної моделі (сертифікат від 02.07.2010).

7. Захищено 1 докторська дисертація (доц. Литвинова Є.І., науковий керівник д.т.н., проф. каф. АПОТ Хаханов В.І.). Впровадження результатів – інфраструктура вбудованого діагностування цифрових систем на кристалах: навчальний процес ХНУРЕ (акт впровадження від 27.11.2009); науково-дослідну та виробничу діяльність ВАТ «АТ НДРВ» (довідка від 12.10.2009); науково-дослідний процес ЗАТ «СНПО «Імпульс»» (акт від 19.11.2009); виробничий процес фірми «Aldec, Inc» (довідка від 09.11.2009).

8. Каменюка Є.О., науковий керівник – д-р техн. наук, проф. Хаханов В.І.; «Мультипроцесорні моделі розв’язання систем булевих рівнянь великої розмірності» (кандидатська, 24.04.2009),

9. Камінська М.О., науковий керівник – д-р техн. наук, проф. Хаханов В.І.; «Системні моделі аналізу тестопридатності при проектування цифрових структур на кристалах» (кандидатська, 23.06.2009),

10. Хаханова Г.В., науковий керівник – проф. Бараннік В.В. «Структурно-каскадні методи стиску та відновлення даних в телекомунікаційних та цифрових системах реального часу» (кандидатська, 13.05.2009),

11. Парфентій О.М., наук. керівник – д.т.н., проф. Хаханов В.І. «Моделі інфраструктури сервісного обслуговування цифрових систем на кристалах» (кандидатська, 02.07.2009).

12. Нгене Кристофер Умерах, наук. керівник – д.т.н., проф. Хаханов В.І. (кандидатська, 14.12.2011). Впровадження результатів дисертації Нгене Кристофера Умераха: 1) Моделі й методи тестування, а також архітектура спеціалізова-

ного мультиматричного процесора доведено до практичної реалізації у вигляді програмно-апаратних компонентів інфраструктури, інтегрованої з системою Riviera (Aldec), що надало можливість створювати ефективні маршрути перевірки та діагностування HDL-коду цифрових систем на кристалах. 2) Інтеграція програмних HDL-моделей і апаратної реалізації методів діагностування функціональних порушень дозволила істотно (20%) скоротити часові витрати для моделювання й верифікації у процесі тестування цифрових виробів. 3) Інфраструктура тестування HDL-коду надає можливість апаратно обробляти програмно-апаратні моделі проєктованого виробу, що дозволяє істотно (50%) зменшити час діагностування та корекції коду в процесі ітераційного налагодження. 4) Запропоновані моделі й методи, що складають основу програмно-апаратної інфраструктури тестування на основі асерцій істотно (20%) підвищують тестопридатність цифрової системи в цілому, що дозволяє зменшити часові витрати на синтез тесту, підвищити його функціональну повноту і якість проєктованого виробу. 5) Обґрунтованість і достовірність наукових положень підтверджується результатами експериментальних досліджень, обробкою 10 реальних моделей систем на кристалах, орієнтованих на промислове використання. Результати експериментів підтверджують високу ефективність діагностування функціональних порушень в HDL-моделях, а так само суттєве підвищення продуктивності програмно-апаратної системи тестування у порівнянні з існуючими академічними аналогами. Достовірність наукових висновків підтверджується інтеграцією розробленої інфраструктури тестування в програмний комплекс Riviera компанії Aldec (США). 6) Розробки у вигляді програмно-апаратних додатків, запропоновані в дисертації, використовуються у: Харківському національному університеті радіоелектроніки, в навчальному процесі (акт про впровадження від 30.05.2011); Компанії Aldec, USA (довідка про впровадження від 11.06.2011).

13) Самі С.А. Механна – наук. керівник – д.т.н., проф. Кривуля Г.Ф. (кандидатська, 14.12.2011). Впровадження результатів дисертації Самі С.А. Механна: створена нечітка експертна діагностична система реального часу, практична ре-

алізація якої дозволяє користувачу поставити діагноз будь-якому функціональному стану комп'ютерної системи чи мережі практично без використання додаткових процедур діагностики. Зменшення кількості продукційних правил при використанні нечіткої логіки дозволяють реалізацію експертної системи з меншими апаратними затратами та з більшою швидкістю. Експертну діагностичну систему доцільно реалізувати як вбудований апаратний пристрій комп'ютерної системи чи мережі. Наукові та практичні результати дисертації використані в ТОВ «Алдек - КТС» (м. Харків) для діагностики технічного стану спеціалізованого комп'ютерного кластеру, що підтверджується довідкою від 1.02.2011р. При впровадженні роботи було досягнуто зниження трудомісткості, скорочення часових витрат на пошук несправності в комп'ютерному кластері, а також підвищення ефективності процедури постановки діагнозу із забезпеченням відповідної глибини пошуку несправності за рахунок використання інтелектуальних засобів діагностування. Наукові положення і результати дисертації використані також на кафедрі АПОТ Харківського національного університету радіоелектроніки при підготовці навчальних курсів з проектування та діагностики комп'ютерних систем і мереж. Виконана на програмному рівні практична реалізація нечіткої експертної діагностичної системи реального часу. Доцільно таку експертну систему реалізувати як вбудовану у вигляді додаткового апаратного пристрою в комп'ютерну систему. Експлуатація такої експертної системи дозволить користувачу поставити діагноз будь-якому функціональному стану комп'ютерної системи чи мережі більш оперативно з меншими витратами.

14) Прийнято до захисту кандидатську дисертацією Ів Тієкура «Векторно-логічна інфраструктура вбудованого тестування цифрових систем на кристалах», науковий керівник – д.т.н., проф. Хаханов В.І.

15) Прийнято до захисту дисертацію Філіпенко І.В. «Моделі та методи проектування систем на кристалах для радіочастотної ідентифікації з кодовим розподілом каналів», науковий керівник – д.т.н., проф. Хаханов В.І.

4.10. Виставки

Результати дослідження були представлені у вигляді експонатів на наступних виставках.

2009 рік:

№ п/п	П.І.Б. розроблювачів експонату	Назва експонату	Назва виставки	Дата й місце проведення виставки	Нагороди
1	Хаханов В.І., Чумаченко С.В.	«Елітна клубна освіта (Club Elite Teaching) – шлях до технологічної культури студентів»	«Інноватика в освіті»	2-3 грудня 2009 р., Київ	Грамота у загальному університетському заліку
2	Хаханов В.І., Чумаченко С.В.	«Елітна клубна освіта (Club Elite Teaching) – шлях до технологічної культури студентів»	«Освіта і кар'єра»	10-11 грудня 2009 р., м. Київ	Грамота у загальному університетському заліку
3	Сушанов О.	Комплекс програмних средств для моделирования, диагностирования и восстановления работоспособности функциональных модулей SIP	«Наука и производство. Машиностроение Харьковщины»	Вересень 2009, Харків	
4	Шкиль О.С., Напрасник С.В., Каук В.І.	Система тестування знань OpenTest2	«Інноватика в освіті»	2-3 грудня 2009 р., Київ	Грамота у загальному університетському заліку
5	Шкиль О.С., Напрасник С.В., Янковський О.А.	Система тестування знань OpenTest2	«Освіта і кар'єра»	10-11 грудня 2009 р., м. Київ	Грамота у загальному університетському заліку
6	Рисухин М.В., Свет личный В.О., Степанов А.И.	Проблема загрязнения окружающей среды отходами промышленности и бытовыми отходами	Каталог выставки технического творчества молодежи 13-й международный молодежный форум «Радиоэлектроника и молодежь в XXI веке»	Харьков, 30 марта-01 апреля 2009 г., с.11-12	

№ п/п	П.І.Б. розроблювачів експонату	Назва експонату	Назва виставки	Дата й місце проведення виставки	Нагороди
7	Хаханов В.І., Чумаченко С.В.	«Елітна клубна освіта (Club Elite Teaching) – шлях до технологічної культури студентів»	«Інноватика в освіті»	2-3 грудня 2009 р., Київ	Грамота у загальному університетському заліку
8	Хаханов В.І., Чумаченко С.В.	«Елітна клубна освіта (Club Elite Teaching) – шлях до технологічної культури студентів»	«Освіта і кар'єра»	10-11 грудня 2009 р., м. Київ	Грамота у загальному університетському заліку
9	Рисухин М.В. (СП-05-1), Светличный В.О. (СП-05-2), Степанов А.И. (СП-05-2)	Проблема загрязнения окружающей среды отходами промышленности и бытовыми отходами	Выставка технического творчества молодежи 13-й международный молодежный форум «Радиоэлектроника и молодежь в XXI веке», Харьков, 30 марта-01 апреля 2009 г., с.11-12	Харьков, 30 марта - 01 апреля 2009 г., с.11-12	
10	Сазонов Я.В. (ИБ-06-3), Сапрыкин А.А. (ИБ-06-3)	“Radiation”	Каталог выставки технического творчества молодежи 13-й международный молодежный форум «Радиоэлектроника и молодежь в XXI веке», Харьков, 30 марта-01 апреля 2009 г., с.13	Харьков, 30 марта-01 апреля 2009 г., с.11-12	
11	Михтонюк С.В. (СКС-05-1), Давыдов М.Д. (СКС-05-1), Першин М.А. (СПм-08-1), Струков М.О. (СУА-05-1), Приходченко Р.С. (КИ-07-6), Кубиря А.В. (СИ-07-2), Мирошниченко С.В. (КИ-07-4)	INTSPEI P-Navigator	Каталог выставки технического творчества молодежи 13-й международный молодежный форум «Радиоэлектроника и молодежь в XXI веке», Харьков, 30 марта-01 апреля 2009 г., с.14-15	Харьков, 30 марта-01 апреля 2009 г., с.11-12	

№ п/п	П.І.Б. розроблювачів експонату	Назва експонату	Назва виставки	Дата й місце проведення виставки	Нагороди
12	Сушанов А.В. (КИ-07-6), Галаган С.С. (СП-05-1)	Программное средство для диагностики SiP	Каталог выставки технического творчества молодежи 13-й международный молодежный форум «Радиоэлектроника и молодежь в XXI веке», Харьков, 30 марта-01 апреля 2009 г., с.15	Харьков, 30 марта-01 апреля 2009 г., с.11-12	
13	Сушанов А.В. (КИ-07-6), Ольховой В. (СКС-06-1)	Программное средство для восстановления работоспособности FPGA	Каталог выставки технического творчества молодежи 13-й международный молодежный форум «Радиоэлектроника и молодежь в XXI веке», Харьков, 30 марта-01 апреля 2009 г., с.15	Харьков, 30 марта-01 апреля 2009 г., с.11-12	
14	Третьяков Н.А. (СКС-06-1)	Программное средство для генерирования исполняемых (Win PE) фаулов, выполняющих заданные пользователем команды	Каталог выставки технического творчества молодежи 13-й международный молодежный форум «Радиоэлектроника и молодежь в XXI веке», Харьков, 30 марта-01 апреля 2009 г., с.16	Харьков, 30 марта-01 апреля 2009 г., с.11-12	
15	Гаврюшенко А. (СКС-05-1), Сухомлинов Д. (СКС-05-1)	Программный продукт для диагностирования ошибок проектирования SOC	Каталог выставки технического творчества молодежи 13-й международный молодежный форум «Радиоэлектроника и молодежь в XXI веке», Харьков, 30 марта-01 апреля 2009 г., с.17	Харьков, 30 марта-01 апреля 2009 г., с.11-12	

№ п/п	П.І.Б. розроблювачів експонату	Назва експонату	Назва виставки	Дата й місце проведення виставки	Нагороди
16	Сліпченко М.І., Єлаков С.Г., Руденко О.Г. Сотников О.М.	Автоматизированная система измерения больших токов	Форум, ХНУРЕ. 8-11 октября 2009		
17	Сліпченко М.І., Єлаков С.Г., Руденко О.Г. Сотников О.М.	Система контроля температурного режима в трубчатой вращающейся печи	Форум, ХНУРЕ. 8-11 октября 2009		
18	Сліпченко М.І., Єлаков С.Г., Руденко О.Г. Сотников О.М.	"Комплекс "Зонд" автоматизированного контроля эффективности текущих электроплавов в рудно-термических электропечах".	Форум ХНУРЭ, 8-11 октября 2009г.		

2010 рік:

П.І.Б. розроблювачів експонату	Назва експонату	Назва виставки	Дата й місце проведення виставки	Нагороди
Бражніков О., студент гр. КІ-08-6, Зайченко С.О., аспірант каф. АПОТ. Науковий керівник: д.т.н., проф. каф. АПОТ Хаханов В.І.	ALINT 2010.06 - система діагностування hdl-проектів	Виставка науково-технічної творчості молоді, присвячена Міжнародному дню студента	Харків, ХНУРЕ, 18.10.1010	2 місто
Короленко А.А., ст. гр. КІ-08-6, Филиппенко И.О., ст.гр. КІ-08-6 Научный руководитель: д.т.н., проф. каф. АПВТ Хаханов В.И.	Стерео УМЗЧ С TFT-дисплеем и сенсорным вводом	Виставка науково-технічної творчості молоді, присвячена Міжнародному дню студента	Харків, ХНУРЕ, 18.10.1010	
Короленко А.А., ст. гр. КІ-08-6, Кокулюк А.А., ст.гр. КІ-08-6 Научный руководитель: д.т.н., проф.	USB программатор микроконтроллеров AVR совместимый с AVR910	Виставка науково-технічної творчості молоді, присвячена Міжнародному дню студента	Харків, ХНУРЕ, 18.10.1010	

П.І.Б. розроблювачів експонату	Назва експонату	Назва виставки	Дата й місце проведення виставки	Нагороди
Хаханов В.И., каф. АПВТ				
Короленко А.А., ст. гр. КИ-08-6 Научный руководитель: д.т.н., проф. каф. АПВТ Хаханов В.И.	USB звуковая аудиокарта	Виставка науково-технічної творчості молоді, присвячена Міжнародному дню студента	Харків, ХНУРЕ, 18.10.1010	
Филиппенко И.О., ст. КИ-08-6 Научный руководитель: д.т.н., проф. каф. АПВТ Хаханов В.И.	Эхо процессор	Виставка науково-технічної творчості молоді, присвячена Міжнародному дню студента	Харків, ХНУРЕ, 18.10.1010	
Филиппенко И.О., ст. КИ-08-6 Научный руководитель: д.т.н., проф. каф. АПВТ Хаханов В.И.	Система цифровой связи	Виставка науково-технічної творчості молоді, присвячена Міжнародному дню студента	Харків, ХНУРЕ, 18.10.1010	
Филиппенко И.О., Короленко А.А., Кокуюк А.А. ст. КИ-08-6 Научный руководитель: д.т.н., проф. каф. АПВТ Хаханов В.И.	Стенд для ознакомления с MP3 форматом	Виставка науково-технічної творчості молоді, присвячена Міжнародному дню студента	Харків, ХНУРЕ, 18.10.1010	
Зацарний А.В., студент гр. КИ-08-6, Мищенко О.С., аспирант каф. АПОТ Научный керівник: Чумаченко С.В., д.т.н., проф. каф. АПОТ	Соціальна мережа www.otdyh.ua	Виставка науково-технічної творчості молоді, присвячена Міжнародному дню студента	Харків, ХНУРЕ, 18.10.1010	
Захарченко О.В., аспирант каф. АПОТ; Бражніков А., Хаханова Ю.В. студенти гр. КИ-08-6 Научный керівник:	Програмный продукт “COVERAGE”	Виставка науково-технічної творчості молоді, присвячена Міжнародному дню студента	Харків, ХНУРЕ, 18.10.1010	

П.І.Б. розроблювачів ек- спонату	Назва експонату	Назва виставки	Дата й місце проведення виставки	Нагороди
Чумаченко С.В., д.т.н., проф. каф. АПОТ				

2011 рік:

П.І.Б. розроблювачів експонату	Назва експонату	Назва виставки	Дата й місце проведення виставки	Нагороди
Короленко А.А., Филиппенко И.О.(гр.КИ-08-6)	Сtereo УМЗЧ с TFT- дисплеем и сенсорным вводом	Виставка XV Міжнародного молодіжного фо- руму «Радіоелектри мо- лодь в XXI ст».	18-20 апреля 2011 года, Харьков.	
Короленко А.А., Кокулюк А.А.(гр.КИ-08-6)	USB программа тор микроконтроллеров AVR совместимый с AVR 910	Виставка XV Міжнародного молодіжного фо- руму «Радіоелектр. и молодь в XXI ст».	18-20 апреля 2011 года, Харьков.	
Короленко А.А. (гр.КИ-08-6)	USB звуковая аудиокар- та	Виставка XV Міжнародного молодіжного фо- руму «Радіоелектр. и молодь в XXI ст».	18-20 апреля 2011 года, Харьков.	
Короленко А.А., Филиппенко И.О. Кокулюк А.А.(гр.КИ-08-6)	Концептуальное устрой- ство для воспроизведе- ния звуковых колебаний разночастотного спектра	Виставка XV Міжнародного молодіжного фо- руму «Радіоелектр. и молодь в XXI ст».	18-20 апреля 2011 года, Харьков.	
Филиппенко И.О.(гр.КИ-08-6)	Эхо процессор	Виставка XV Міжнародного молодіжного фо- руму «Радіоелектр. и молодь в XXI ст».	18-20 апреля 2011 года, Харьков.	
Филиппенко И.О.(гр.КИ-08-6)	Система цифровой связи	Виставка XV Міжнародного молодіжного фо- руму «Радіоелектр. и молодь в XXI	18-20 апреля 2011 года, Харьков.	

		ст».		
Филиппенко И.О., Короленко А.А., Кокулюк А.А.(гр.КИ-08-6)	Стенд для ознакомлення с MP3 форматом	Виставка XV Міжнародного молодіжного фо- руму «Радіоелектр. и молодь в XXI ст».	18-20 апреля 2011 года, Харьков.	
Филиппенко И.О., Короленко А.А., Кокулюк А.А.(гр.КИ-08-6)	MP3 плеер с повышен- ным качеством звучания и поддержкой плагинов	Виставка XV Міжнародного молодіжного фо- руму «Радіоелектр. и молодь в XXI ст».	18-20 апреля 2011 года, Харьков.	

4.12. Відкриття лабораторій

У 2009 р. створено спеціалізовану науково-дослідницьку лабораторію web-технологій та ІТ-інновацій – «DataArt Lab».

Науковим напрямком Лабораторії є проблема створення нових сервісів та інфраструктури кіберпростору.

Мета Лабораторії – стандартизація кіберпростору та створення ефективних двигунів для точного пошуку інформації.

Напрямки роботи Лабораторії пов'язані із з впровадженням у світову практику сучасних науково-технічних та інноваційних технологій, що стосуються останніх досягнень з ІТ-топ десятки від Gartner-груп (на 2009 р.): 1) Віртуалізація; 2) «Хмарові обчислення» (cloud computing); 3) Сервери майбутнього, що прийдуть на зміну blade-серверів; 4) Web-орієнтовані архітектури; 5) Змішані Корпоративні додатки (mashups); 6) Спеціалізовані системи; 7) Соціальні Мережі та Програмне забезпечення для них; 8) Об'єднані комунікації (unified communications); 9) Бізнес-Аналітика (BI); 10) «Зелені» ІТ.