

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Тернопільський національний економічний університет  
Навчально-науковий інститут інноваційних освітніх технологій  
Кафедра комп'ютерної інженерії

**ПАК Святослав Іванович**

**Моделі та структури нейроелемента з паралельно-  
груповим обчисленням / Models and structures of a  
neuroelement with parallel-group calculation.**

спеціальність: 123 - Комп'ютерна інженерія  
магістерська програма - Комп'ютерна інженерія

Магістерська робота

Виконав студент групи КІм-21  
С.І. Пак  
Науковий керівник:  
д.т.н., професор, І. Г. Цмоць

Магістерську роботу допущено до захисту:

ТЕРНОПІЛЬ -2018

## ЗМІСТ

Вступ.....	9
1. Аналіз галузей застосування і архітектур нейромереж.....	10
1.1. Галузі застосування нейромереж .....	10
1.2. Архітектури штучних нейромереж .....	17
1.3. Засоби та варіанти реалізації нейроалгоритмів .....	25
1.4. Постановка задачі магістерської роботи .....	28
2 Алгоритми реалізації базових нейрооперацій.....	29
2.1 Вибір принципів та формування вимог до компонентів комп'ютерних штучних нейронних мереж реального часу .....	29
2.2 Паралельно-вертикальні алгоритми та структури пристроїв групового підсумовування .....	29
2.3. Паралельно-вертикальні алгоритми та НВІС-структури пристроїв обчислення суми квадратів різниць .....	39
2.4 Паралельно-вертикальні алгоритми та структури пристроїв обчислення максимальних і мінімальних значень.....	43
3 Вертикально-паралельні методи обчислення та синтез апаратних компонентів штучних нейронних мереж .....	50
3.1 Методи та структури апаратних компонентів для реалізації скалярного добутку .....	50
3.2 Методи та структури апаратних компонентів для реалізації групового підсумовування .....	71
3.3 Метод та структура апаратного компонента для реалізації пристрою визначення максимального числа з групи чисел .....	81
3.4 Метод та апаратний компонент реалізації пристрою для обчислення суми квадратів різниць .....	85
Висновки .....	92

Список використаних джерел ..... 94

ДОДАТОК А СВІТЛОКОПІЯ ПУБЛІКАЦІЇ **Ошибка! Закладка не определена.**

ДОДАТОК Б ДОВІДКА ПРО ВИКОРИСТАННЯ **Ошибка! Закладка не определена.**

## ВСТУП

Актуальність теми. Сучасний етап розвитку штучних нейромережових (ШНМ) технологій характеризується розширенням галузей застосування, в значній частині з яких вимагається опрацювання за складними алгоритмами у реальному часі різних за інтенсивністю надходження потоків даних на апаратних засобах, що задовільняють обмеження щодо габаритів, енергоспоживання, вартості та часу розробки. Створення таких апаратних засобів вимагає широкого використання сучасної елементної бази (напівзамовних і замовних НВІС, однокристальних мікропроцесорів) та розробки нових методів, алгоритмів та НВІС-структур для реалізації багатооперандних нейрооперацій.

Аналіз задач, що розв'язуються за допомогою ШНМ, методів і алгоритмів їх розв'язання виявив такі особливості: великий об'єм обчислень з перевагою обчислювальних операцій над логічними; постійність і висока інтенсивність надходження даних; регулярність і рекурсивність алгоритмів; можливість розпаралелювання процесу обробки як у часі, так і в просторі.

Для забезпечення розв'язання задач в реальному часі за допомогою ШНМ повинна бути висока продуктивність апаратних засобів. Режим реального часу при обробці інтенсивних потоків даних за складними алгоритмами забезпечується розпаралелюванням і конвеєризацією процесів обчислень та використанням нових технологічних досягнень в області розробки надвеликих інтегральних схем (НВІС). Мета і задачі дослідження. Метою магістерської роботи є розробка засобів для реалізації багатооперандних нейрооперацій.

Для досягнення поставленої мети в роботі розв'язуються такі задачі:

- аналіз галузей застосування і архітектур нейромереж
- розробка алгоритмів реалізації базових нейрооперацій
- розробка апаратно-програмних засобів, що реалізують компоненти нейромережових технологій

# 1. АНАЛІЗ ГАЛУЗЕЙ ЗАСТОСУВАННЯ І АРХІТЕКТУР НЕЙРОМЕРЕЖ

## 1.1. Галузі застосування нейромереж

Задачі, які розв'язуються за допомогою нейромережових технологій, можна об'єднати у такі типові групи: функціональне моделювання, класифікація, динамічна фільтрація, прогнозування, ідентифікація систем, кластеризація, виділення ознак та аномалій [1].

Функціональне моделювання. Функціональне моделювання полягає у визначенні цифрового виходу в залежності від набору цифрових входів. Нейромережа створює модель, яка знаходить зв'язок між входами та потрібним виходом. Таке моделювання здатне розв'язати велике коло задач. Прикладом може бути оцінка цін на житло (вихід), у залежності від ряду показників – входів (розмір будинку, розмір земельної ділянки, відстань до найближчої школи тощо) [2].

Ще одним прикладом може бути навчання нейронної мережі керування автомобілем, де входом є зображення дороги попереду, а виходом – сигнали управління кермом, педалями газу, гальма тощо.

Функціональне моделювання охоплює надзвичайно широке коло сфер застосування, і нейромережі найчастіше використовуються саме для цього. Його можна розглядати як альтернативу традиційному програмуванню. При традиційному програмуванні комп'ютеру безпосередньо вказується за допомогою інструкцій, як саме виконувати обчислення або задачу. При функціональному моделюванні, замість того, щоб вказувати комп'ютеру як щось робити, подаються приклади того, що потрібно зробити. При цьому надаються дані про входні значення і показується, які вихідні значення при цьому повинні бути, а нейронна мережа вже самостійно моделює функцію залежності виходу від входів.

Класифікація є окремим випадком функціонального моделювання, при якому здійснюється розпізнавання образів. Нейронна мережа навчається як

категоризувати вхідні значення, які вона отримує, у набір класів. Нейронні мережі показали перевагу над традиційними методами у всіх аспектах класифікації і дуже популярні у сфері розпізнавання образів.

Типовим прикладом класифікації є розпізнавання рукописного тексту. На вхід такої системи подається рукописний символ, а вихід показує який саме це символ (конкретна буква або цифра) [3].

Іншим прикладом може бути тестування якості пива (яке дійсно використовується деякими пивоварнями), при якому нейронна мережа розподіляє різні партії пива у різні класи якості (вихід), у залежності від різних хімічних показників (входи) [4].

Класифікація також є наслідком парадигми навчання на прикладах. Нейронна мережа навчається на прикладах вхідних даних та їхньої належності до певного класу. При цьому нейронна мережа будує генеральний класифікатор, який може використовуватися на нових вхідних даних.

Динамічна фільтрація. Описані вище задачі мали справу зі статичними даними, тобто, коли один конкретний вхід призводив до одного конкретного виходу. У системах, які мають пам'ять (де вихідні значення залежать від попередніх вхідних значень) і внутрішню динаміку, виникають проблеми різних типів. Такі задачі пов'язані з тимчасовою динамікою і їх дуже складно розв'язати традиційними алгоритмічними методами. При цьому нейронні мережі без проблем справляються з поставленим завданням.

Прикладом задачі динамічної фільтрації є нейромережне придушення шумів, яке можна використати для дуже ефективного видалення шумів з різноманітних (наприклад, звукових) сигналів [5].

Іншим прикладом задачі динамічної фільтрації є класифікація динамічних сигналів. На відміну від звичайної класифікації, яка відображає набір входів у клас, динамічна фільтрація класифікує зміну сигналу у часі. Прикладом може бути розпізнавання руху тіла, при якому людина однозначно ідентифікується на основі того, як вона рухається .

Динамічна фільтрація, як і класифікація, також використовує принцип навчання на прикладах. Різниця між ними полягає у тому, що входами тут є цілі послідовності вхідних значень.

Прогнозування. Прогнозування є особливим типом динамічної фільтрації, у якому попередні значення використовуються для прогнозування майбутніх значень. Воно широко використовується для передбачення у випадках, коли є часовий ряд, який простягається до теперішнього часу, і потрібно спрогнозувати, яким він буде у майбутньому [5].

Одним з типових прикладів прогнозування є передбачення стану фондового ринку, або ринку Forex. Іншим прикладом прогнозування є передбачення піків навантаження у електромережі .

Передбачення майбутніх станів дуже складних систем є нетривіальним завданням. Але нейронні мережі, що були навчені на правильних навчальних прикладах, показали, що цілком здатні виробляти добрі прогнози. І вони показують набагато кращі результати, ніж більш традиційні методи прогнозування.

Ідентифікація систем. Ідентифікація систем є ще одним спеціальним типом динамічної фільтрації, метою якого є побудова математичної моделі, що відображає функції та динаміку реальної системи. Такі моделі є надзвичайно важливими при аналізі, симуляції, передбаченні, моніторингу та розробки системи управління. Найбільшою відмінністю між нейронними мережами та традиційними методами ідентифікації систем є те, що нейронні мережі є нелінійними, що дає їм набагато більші можливості моделювання.

Іншою великою перевагою є те, що нейронні мережі можуть використовувати пасивне дослідження для створення моделі. Традиційні методи зазвичай потребують явного визначення спектральних властивостей системи за допомогою надсилання білого шуму через неї. Можна зауважити, що ніхто насправді не захоче надсилати білий шум через атомну електростанцію. А нейронні мережі можуть «слухати» тільки нормальні сигнали, без небажаних досліджень, і при цьому все-одно будувати добрі загальні моделі.

Типовим прикладом доцільності застосування нейромережної ідентифікації систем є галузь обробної промисловості. У ній важливо мати хорошу модель заводу (наприклад, хімічного заводу), з метою виявлення аномалій, виконання оптимізації, тощо.

Результат ідентифікації систем може бути поданий на вхід, формуючи при цьому інверсивну ідентифікацію системи, тобто, систему управління. У такому режимі нейронній мережі вказується, яким повинен бути результат діяльності заводу, а вона показує, які вхідні дані потрібно при цьому забезпечити [6].

Кластеризація. Кластеризація – це розподіл множини вхідних даних на групи (кластери) за рівнем подібності між собою. Нейронні мережі можуть виконувати автономну кластеризацію даних, тобто, не потрібно забезпечувати будь-яких сигналів зворотного зв'язку чи вказувати системі що потрібно робити [7, 8].

Зокрема, кластеризація застосовується для визначення сегментації ринку. За допомогою основних даних опитування нейронна мережа може згрупувати людей за подібністю (багатовимірною). Системи кластеризації є дуже корисними інструментами для аналізу даних (data mining).

Виділення ознак. Виявлення ознак – це задача знаходження характерних рис даних. Нейронні мережі можуть виконувати автономне виділення ознак з вхідних даних. Ця властивість є дуже цінною для стиснення даних. Оскільки нейронні мережі адаптуються до конкретних даних, то вони можуть зробити набагато краще стиснення, ніж певний універсальний метод стиснення.

Виявлення аномалій. Нейронні мережі можуть вивчити стандартну поведінку системи (або людини) і можуть певним чином реагувати, коли ця поведінка стає нехарактерною. Цей процес називається виявленням аномалій, або ж фільтром новизни.

Типовим прикладом може бути система мережевої безпеки, у якій мережевий трафік комп'ютера контролюється і дані про активність надсилаються до адаптивної системи. Через деякий час адаптивна система



створює модель нормальної роботи мережі. Коли відбувається щось незвичайне (наприклад, спроба хакерської атаки), адаптивна система повідомляє про це.

Виявлення аномалій є важливим на промислових підприємствах та у медичному обладнанні для моніторингу. Воно також може використовуватися як індикатор стану фінансового ринку.

Успіхи в теорії штучних нейронних мереж (ШНМ), інтенсивний розвиток НВІС-технологій та комп'ютерних засобів істотно розширили галузі застосування нейромережевих технологій. Серед багатьох галузей застосування ШНМ розглянемо найважливіші, до яких можна віднести фінанси, медицину, промисловість, споживчі товари та інформаційні технології [9] (табл. 1.1).

Таблиця 1.1. Основні галузі застосування нейромережевих технологій

№ п/п	Галузь застосування	Задачі, що розв'язуються в галузі
1.	2.	3.
1.	Фінанси	<ul style="list-style-type: none"> <li>— фінансове прогнозування [10] – нейронні мережі є дуже популярними для нелінійного прогнозування фінансових даних, таких як передбачення фондового ринку. Вони зарекомендували себе як дуже потужний інструмент, який набагато перевершує всі традиційні методи. Більшість великих компаній, які торгують на фондовому ринку, використовують ті чи інші адаптивні системи, зокрема, нейронні мережі, для прогнозування стану ринку;</li> <li>— виявлення шахрайства [11] – банки та інші фінансові установи використовують системи на основі нейронних мереж для виявлення шахрайства (кредитні картки, транзакції тощо). Нейронна мережа вивчає звички та стиль власника рахунку, і це</li> </ul>

## Продовження таблиці 1.1

1.	2.	3.
2.	Промисловість	<ul style="list-style-type: none"> <li>— моделювання та контроль заводів [16] – одним з найуспішніших та найпопулярніших застосувань нейронних мереж є моделювання та контроль заводів у переробній промисловості. Нейронні мережі є набагато дешевшою, надійнішою та стійкішою альтернативою традиційному фізичному моделюванню роботи заводів. Їх успішно впроваджено у багатьох галузях переробної промисловості (целюлозно-паперовій, хімічній тощо);</li> <li>— вирівнювання навантаження в електромережах [17] – сучасні системи вирівнювання навантаження в електромережах використовують нейронні мережі для оптимізації та регуляції своєї роботи;</li> <li>— робототехніка [18] – управління роботами було однією з найперших областей застосування нейромереж. Тут вони широко використовуються для виконання найрізноманітніших задач, таких як організація штучного зору, автономне керування рухом тощо;</li> <li>— автомобільна промисловість [19] – сучасні автомобілі використовують безліч різних вбудованих систем, починаючи від системи контролю тяги і до контролера регулювання сидіння. Майже у кожній з цих систем нейронні мережі мають значні переваги у порівнянні з традиційним програмуванням.</li> </ul>
3.	Споживчі товари	<ul style="list-style-type: none"> <li>— пральні машини [7] – більшість сучасних пральних машин містять схеми нечіткої логіки та особливі типи нейронних мереж, які використовуються для оптимізації витрат води, прального порошку тощо, у залежності від різних показників датчика одягу всередині пральної машини;</li> <li>— оптичні миші [9] – у оптичних мишах використовується спеціальний тип заздалегідь навчених нейронних мереж для виявлення руху. Ці нейронні мережі є типом фільтрів новизни, який виявляє аномалії відносної рівноваги (вхідними даними є зображення з CCD-датчика оптичної миші);</li> <li>— тостери [9] – у більшості сучасних тостерів нейромережі використовуються для керування процесом приготування тостів на основі показників, отриманих з датчика вставленого хліба;</li> <li>— струменеві принтери [6] – струменеві принтери використовують нейронні мережі для регулювання потоку чорнила на головку принтера на основі типу паперу, типу друку і т.д.;</li> <li>— мікрохвильові печі [2] – нейронні мережі використовуються у складних мікрохвильових печах для регулювання вихідної потужності на основі різних показників датчика їжі;</li> <li>— цифрові фотоапарати цифрові фотоапарати використовують нейронні мережі для налаштування різних параметрів залежно від умов навколишнього середовища.</li> </ul>

## Продовження таблиці 1.1

1.	2.	3.
4.	Медицина	<p>— медична діагностика [2] – нейронні мережі використовуються для діагностики раку та інших захворювань. Наприклад, вони успішно використовуються для дуже точної діагностики раку молочної залози. При цьому вхідними даними є вимірювання з клітинних зразків. Також вони успішно та точно діагностують ураження печінки, цукровий діабет, серцеві захворювання тощо;</p> <p>— біоінформатика – штучні нейронні мережі успішно використовуються у біоінформатиці. Зокрема, їх використовують для передбачення вторинної структури білків, ідентифікації генів, моделювання білково-білкової взаємодії, відкриття шляхів метаболізму, мікроматричного аналізу та розпізнавання центрів зв'язування ДНК;</p> <p>— прогнозування взаємодії ліків – нейронні мережі можуть використовуватися для прогнозування можливих побічних ефектів при взаємодії різних лікарських засобів. Вони також можуть використовуватися для вивчення природи алергічних реакцій та шляхів їх виникнення (часто до алергічної реакції призводить взаємодія декількох різних речовин);</p> <p>— оцінка ймовірності виживання – нейронні мережі можуть використовуватися для встановлення черговості медичної допомоги на основі даних про пацієнтів. Вони забезпечують високу точність прогнозу, і в той же час звільняють лікаря від прийняття важкого рішення про вибір, кого саме лікувати чи кому здійснювати трансплантацію органів. У Великобританії існує національна експериментальна програма, що використовує нейронні мережі, які на основі 80 різних вимірних факторів визначають, кому саме робити пересадку печінки;</p>
5.	Інформаційні технології	<p>— безпека – нейронні мережі дуже популярні в системах виявлення вторгнень, які навчаються розрізняти звичайний мережний трафік та спроби вторгнення зловмисників. При цьому нейронна мережа вивчає характеристики звичайного трафіку і попереджає про аномальну активність у роботі мережі;</p> <p>— пошукові системи – велика кількість різних видів нейронних мереж використовуються у пошукових системах для семантичного аналізу та виділення признаков тексту;</p> <p>— розпізнавання образів – нейронні мережі у даній області повністю перевершують традиційні методи. Тому вони використовуються у системах комп'ютерного зору, біометрії, при розпізнаванні мовних сигналів та рукописного тексту і т.д.;</p> <p>— комп'ютерні ігри – нейронні мережі дуже популярні при створенні штучного інтелекту для ігрових персонажів.</p>

Аналіз показав, що значна частина застосувань передбачає опрацювання інтенсивних потоків даних у реальному часі засобами, які задовольняють обмеження щодо габаритів, енергоспоживання та вартості. Зокрема, до таких галузей належать військова промисловість, енергетика, автомобілебудування, управління технологічними процесами, складними об'єктами, мобільними інтелектуальними робототехнічними системами. Зазначено, що найхарактернішими особливостями задач та нейротехнологій реального часу є: постійність і висока інтенсивність надходження вхідних даних; великий обсяг обчислень з переважанням обчислювальних операцій над логічними; регулярність і рекурсивність нейромережевих алгоритмів опрацювання даних; постійне ускладнення алгоритмів опрацювання та підвищення вимог до точності результатів; можливість розпаралелення опрацювання як у часі, так і у просторі; здатність до узагальнення та абстрагування; навчання, самонавчання та самоорганізація під впливом зовнішнього середовища.

## 1.2. Архітектури штучних нейромереж

Штучну нейромережу можна розглядати як направлений граф зі зваженими зв'язками, у якому штучні нейрони є вузлами [20]. У залежності від типу нейронів, які використовуються у нейромережах, а також від вигляду зв'язків між ними розрізняються різні види архітектур нейромереж. Далі розглянемо найпоширеніші з них.

Найпоширенішою архітектурою нейромережі є багат шаровий перцептрон (БШП) [19]. Дані нейромережі належать до мереж прямого поширення (граф мережі не має петель), а нейрони у них розміщені шарами і мають однонаправлені зв'язки між шарами. Ці мережі є статичними, тобто, на заданий

вхід вони виробляють одну сукупність вихідних даних, яка не залежить від попереднього стану мережі.

Багатошарові перцептрони складаються з шару вхідних нейронів, одного або кількох прихованих шарів та шару вихідних нейронів. Нейрони вхідного шару просто ретранслюють сигнали на прихований шар, не змінюючи їх. У прихованих та вихідному шарах відбувається перетворення сигналів та формування реакції мережі. Кількість нейронів в одному шарі визначає розмір шару. Використання прихованих шарів дозволяє нейромережі здійснювати нелінійне перетворення вхід-вихід будь-якої складності. На рисунку 1.1 подано структуру тришарового перцептрона з одним прихованим шаром. У БШП навчання відбувається за алгоритмами навчання з вчителем, наприклад, зворотного поширення похибки.

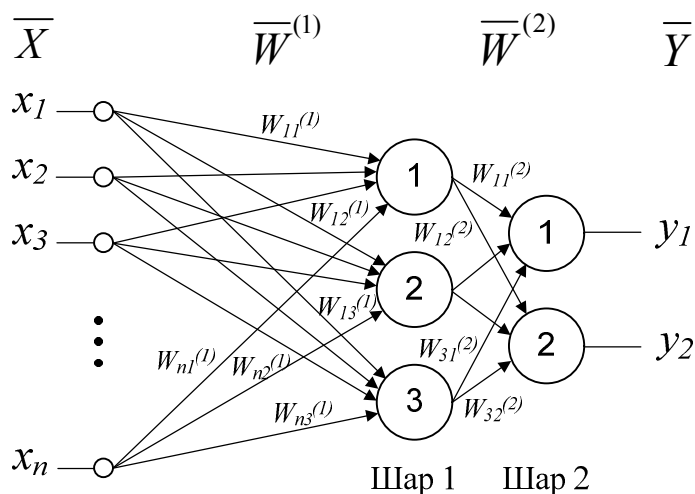


Рисунок 1.1 - Багатошаровий перцептрон.

RBF-мережа також належить до мереж прямого поширення. Ця мережа має прихований шар з радіальних елементів, кожен з яких відтворює гаусову поверхню відгуку [6] (рисунок. 1.2). Оскільки ці функції є нелінійними, то для моделювання довільної функції достатньо використати один прихований шар. Для моделювання будь-якої функції просто потрібно взяти достатню кількість

радіальних елементів і вирішити, яким чином потрібно скомбінувати виходи прихованих радіальних елементів, щоб отримати правильний вихід мережі.

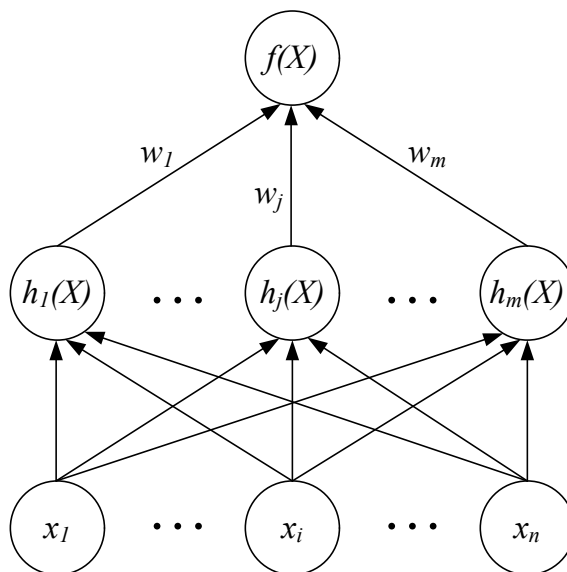


Рисунок 1.2 - RBF-мережа.

Порівняно з БШП RBF-мережа має ряд переваг, зокрема, вони можуть моделювати довільну функцію за допомогою одного прихованого шару, а параметри лінійної комбінації у вихідному шарі можна повністю оптимізувати за допомогою методів лінійного моделювання, які є швидкими і не мають проблем з локальними мінімумами. Недоліком є те, що вони погано придатні для розв'язання задач з великою розмірністю вхідних даних.

При вирішенні задач класифікації можна оцінити щільність ймовірності для кожного класу, порівняти між собою ймовірності належності різним класам і вибрати найбільш ймовірний. Саме це і відбувається, коли ми навчаємо нейронну мережу вирішувати задачу класифікації – мережа пробує визначити щільність ймовірності. Класичний підхід до задачі полягає в тому, щоб побудувати оцінку для щільності ймовірності за відомими даними. Інший підхід до оцінки щільності ймовірності побудований на ядерних оцінках [8]. Можна розмірковувати так: той факт, що спостереження розміщене у даній точці простору свідчить про те, що в цій точці є деяка щільність ймовірності. В методі

ядерних оцінок у точці, що відповідає кожному спостереженню, розміщується деяка проста функція (найчастіше – гаусова функція), потім всі вони додаються і в результаті виходить оцінка для загальної щільності ймовірності.

Метод апроксимації щільності ймовірності за допомогою ядерних функцій багато в чому подібний на метод радіальних базисних функцій, і таким чином ми приходимо до поняття ймовірнісної нейронної мережі (Probabilistic Neural Network, PNN) [6] і узагальнено-регресійної нейронної мережі (GRNN) [67]. PNN-мережа (Рисунок. 1.3) призначена для задач класифікації, а GRNN (Рисунок. 1.10) – для задач регресії. Мережі цих двох типів являють собою реалізацію методів ядерної апроксимації, що оформлені у вигляді нейронної мережі.

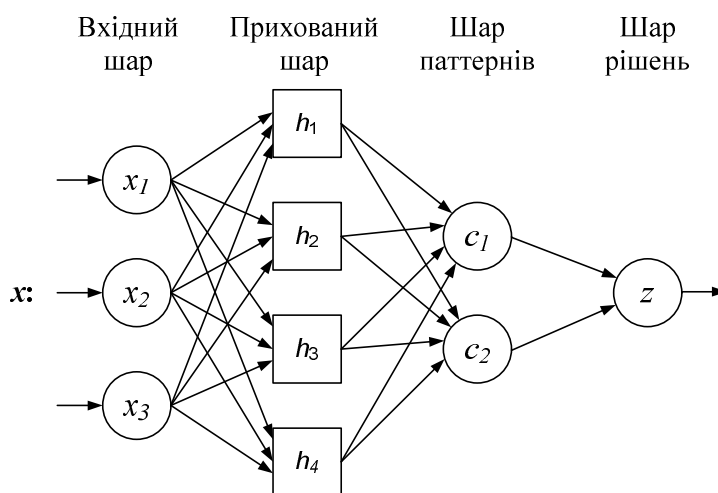


Рисунок 1.3 - Ймовірнісна нейронна мережа.

Мережа PNN має щонайменше три прошарки: вхідний, радіальний та вихідний. Радіальні елементи беруться по одному на кожне навчальне спостереження. Кожен з них представляє гаусову функцію з центром в цьому спостереженні. Кожному класу відповідає один вихідний елемент. Кожен такий елемент з'єднаний з усіма радіальними елементами, що відносяться до його класу, а з усіма іншими радіальними елементами він має нульове з'єднання. Таким чином, вихідний елемент просто додає відгуки всіх елементів, що

належать до його класу. Значення вихідних сигналів виходять пропорційними до ядерних оцінок ймовірності приналежності відповідним класам, і пронормувавши їх на одиницю, ми отримуємо остаточні оцінки ймовірності належності класам.

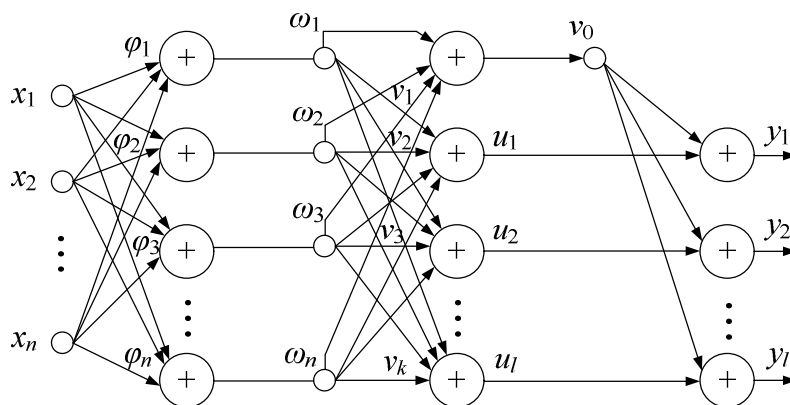


Рисунок 1.4 - Узагальнена регресійна нейронна мережа.

Узагальнено-регресійна нейронна мережа (General Regression Neural Network, GRNN) влаштована аналогічно до ймовірнісної нейронної мережі (PNN), але вона призначена для вирішення завдань регресії, а не класифікації [6-8]. Як і у випадку PNN-мережі, в точку розташування кожного навчального спостереження поміщається гаусова ядерна функція. Будемо вважати, що кожне спостереження свідчить про певну міру впевненості у тому, що поверхня відгуку в даній точці має певну висоту, і ця впевненість зменшується при відході в сторону від точки. GRNN-мережа копіює всередину себе всі навчальні спостереження і використовує їх для оцінки відгуку в довільній точці. Остаточна вихідна оцінка мережі отримується як зважене середнє виходів по всіх навчальних спостереженнях, де величини ваг відображають відстань від цих спостережень до тієї точки, в якій проводиться оцінювання (і, таким чином, більш близькі точки вносять більший внесок в оцінку).

Перший проміжний шар мережі GRNN складається з радіальних елементів. Другий проміжний шар містить елементи, які допомагають оцінити



зважене середнє. Для цього використовується спеціальна процедура. Кожен вихід має в цьому шарі свій елемент, що формує для нього зважену суму. Щоб одержати зі зваженої суми зважене середнє, цю суму потрібно поділити на суму вагових коефіцієнтів. Останню суму обчислює спеціальний елемент другого шару. Після цього у вихідному шарі відбувається власне поділ (за допомогою спеціальних елементів «ділення»). Таким чином, число елементів в другому проміжному шарі на одиницю більше, ніж у вихідному шарі. Як правило, в задачах регресії потрібно оцінити одне вихідне значення, і, відповідно, другий проміжний шар містить два елементи.

Мережі Кохонена принципово відрізняються від всіх інших типів мереж, описаних перед цим. Всі інші мережі призначені для задач з керованим навчанням, а мережі Кохонена головним чином розраховані на некероване навчання [7].

Одне з можливих застосувань таких мереж – розвідувальний аналіз даних. Мережа Кохонена може розпізнавати кластери в даних, а також встановлювати близькість класів. Таким чином, користувач може поліпшити своє розуміння структури даних, щоб потім уточнити нейромережеву модель. Якщо в даних розпізнані класи, то їх можна позначити, після чого мережа зможе вирішувати завдання класифікації. Мережі Кохонена можна використовувати і в тих задачах класифікації, де класи вже задані, – тоді перевага буде в тому, що мережа зможе виявити подібність між різними класами.

Мережа Кохонена має всього два прошарки (рисунок 1.6): вхідний і вихідний, складений з радіальних елементів (вихідний шар називають також шаром топологічної карти). Елементи топологічної карти розташовуються в деякому просторі – зазвичай, двовимірному.

Аналіз класичних нейромережевих методів, алгоритмів і архітектур показав, що вони мають такі недоліки: значний час, необхідний для процесу навчання нейромереж; різні алгоритми, які використовуються для навчання та функціонування нейромереж; архітектури нейромереж не орієнтовані на НВІС-реалізацію. Позбавлені більшості цих недоліків нейромережі на основі моделі

геометричних перетворень (МГП), які відомі під назвою функціонал на множині табличних функцій.

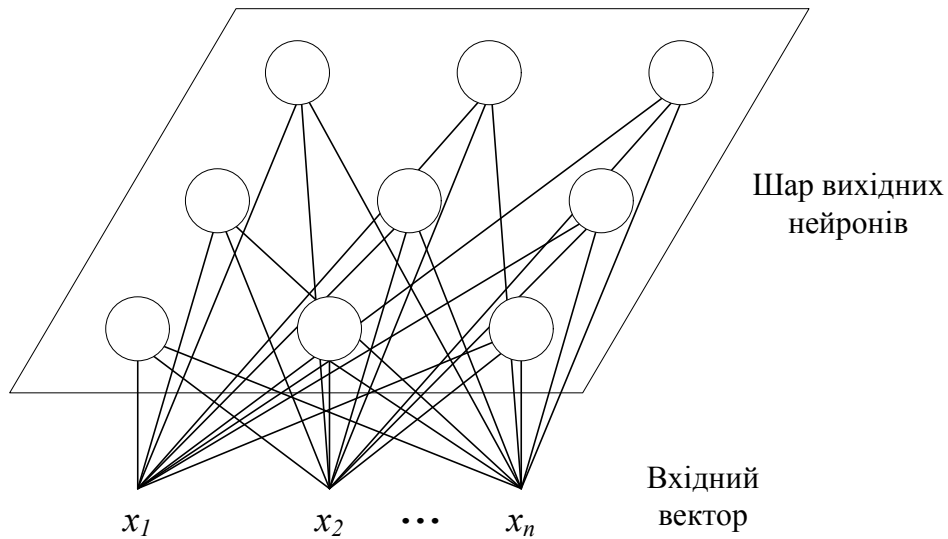


Рисунок 1.5 - Мережа Кохонена.

Використання даних методів і алгоритмів забезпечує суттєве підвищення продуктивності як для програмного, так і апаратного варіантів реалізації нейромереж. Особливістю НС на базі ГП є однотипність алгоритмів навчання та функціонування. Базову структуру нейромережі на основі МГП подано на рисунку 1.6.

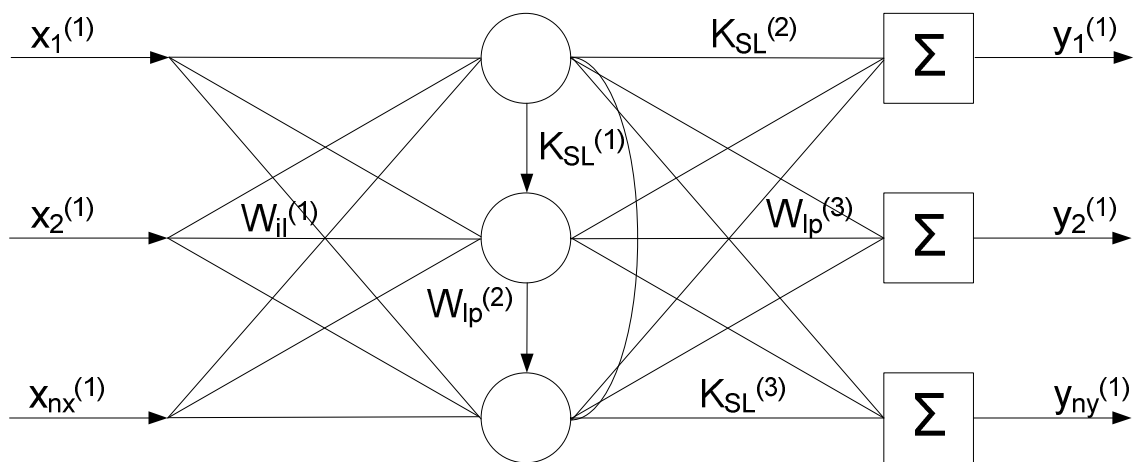


Рисунок 1.6 - Мережа на основі моделі геометричних перетворень.

В основі парадигми МГП покладений принцип представлення гіперповерхонь відгуків в ортогональних системах координат, які максимально збігаються з основними вимірами гіперповерхонь. Близьким аналогом нейромережі на основі парадигми МГП є двошаровий перцептрон автоасоціативного типу, який побудований за методом «звуженого горла». У загальному випадку «звуженого горла», коли число нейронних елементів прихованого шару менше за число входів, перетворення вхідних векторів у ідентичні їм вихідні відбувається з певною похибкою. Перевага мережі МГП полягає у тому, що для її реалізації режим «звуженого горла» не є обов'язковим, отже, існує можливість точного (з нульовою методичною похибкою) відображення векторів вхідних сигналів у вектори вихідних, одночасно виділяючи на виходах нейронних елементів прихованого шару сигнали всіх компонентів інформаційного об'єкта .

На входи мережі МГП подаються одночасно всі компоненти наявних векторів вибірки, ці ж компоненти повторюються як вихідні сигнали тренувальних векторів мережі для здійснення навчання. Вихідні сигнали нейронних елементів відображають сигнали головних компонентів. Детально алгоритм навчання та функціонування мережі МГП розглянуто у [15].

У мережі МГП гіперповерхні відгуку є гіперплощинами, додатковий вимір моделі повністю визначається шумовими компонентами та похибками заокруглень. Результати застосування МГП – основні виміри гіперплощини – збігаються з результатами, отриманими за допомогою відомих методів аналізу головних компонент. Крім цього, застосування мережі МГП надає низку переваг, зокрема цей метод швидкий, неітеративний, без накопичення похибок і помітних обмежень на вимірність, відпадає потреба в розв'язках систем нормальних рівнянь, або у здійсненні ітеративної адаптації.

Основними властивостями НС на базі ГП є наступні: єдина методологічна основа побудови архітектури для різних завдань та галузей застосування; швидке неітеративне навчання за наперед задану кількість кроків обчислень, що відкриває можливість розв'язування завдань великих розмірностей; повна

повторюваність результатів навчання; можливість отримання задовільних розв'язків для тренувальних вибірок зменшеного об'єму; можливість розв'язування задач в автоматичному режимі; розширення кібернетичного принципу “чорної скриньки” на користь “сірої скриньки”, оскільки НС на базі ГП володіють додатковими можливостями аналізу внутрішньої структури даних; висока точність та покращені генералізуючі властивості.

### 1.3. Засоби та варіанти реалізації нейроалгоритмів

Особливістю розвитку нейрокомп'ютерних систем є вдосконалення процесорної компоненти в напрямку збільшення її гнучкості (адаптивності). Ефективність комп'ютерної реалізації нейроалгоритмів безпосередньо пов'язана з вибором засобів реалізації: програмних, мікропрограмних або апаратних [20].

Програмна реалізація передбачає використання універсальних і функціонально-орієнтованих мікропроцесорів для синтезу нейрокомп'ютерних систем. При програмній реалізації нейроалгоритмів обчислювальні процеси переважно розгортаються в часі з великим об'ємом пересилання інформації між оперативною пам'яттю і операційними пристроями. Програмні засоби є доступними для програміста, перед яким виникає задача мінімізації об'єму програм і часу їх реалізації при заданій точності обчислень. Вказані засоби характеризуються низькою швидкістю і гнучкістю з точки зору можливості модифікації та заміни алгоритмів.

Головною особливістю мікропрограмних засобів є регулярна структура, у склад якої входить пам'ять мікропрограм реалізації нейроалгоритмів. Мікропрограмна реалізація обчислень передбачає їхнє розгортання як у часі, так і в просторі. Реалізація в повній мірі потенційних можливостей мікропрограмних засобів може бути досягнута лише при глибокому вивченні як задачі, яка

розв'язується, так і внутрішньої мови процесора. Мікропрограмні засоби реалізації нейроалгоритмів в порівнянні з програмними є більш швидкодіючими.

Успіхи в області інтегральної технології дозволяють все більше перекладати реалізацію нейроалгоритмів на апаратні засоби, які розгортають обчислення не тільки у часі, а і в просторі. Такі обчислення характеризуються введенням додаткового обладнання і відсутністю проміжних пересилок інформації в процесі обчислення, а також спрощенням функції місцевого управління. В основі структурної організації апаратних засобів лежить принцип адекватного апаратного відображення графів алгоритмів функціонування штучних нейронних мереж (ШНМ) на процесорні елементи які реалізують функції нейронів та з'єднані між собою відповідно з графом.

Потрібно відзначити, що всі види реалізації нейроалгоритмів в безпосередньому вигляді зустрічаються досить рідко. На практиці в більшості випадків для реалізації алгоритмів функціонування ШНМ використовуються комбіновані підходи з перевагою одного з перерахованих засобів.

Виходячи із аналізу засобів реалізації нейроалгоритмів, можливі такі варіанти побудови нейрокомп'ютерних систем:

- на основі універсальних і функціонально-орієнтованих мікропроцесорів шляхом розробки спеціалізованого програмного забезпечення;
- на основі універсального обчислювального ядра доповненого базовими апаратно-програмними компонентами, які реалізують часомісткі алгоритми функціонування ШНМ;
- у вигляді спеціалізованої алгоритмічної системи, архітектура та організація обчислювального процесу в якій відображає структуру алгоритму функціонування ШНМ.

Перший варіант є доступним для широкого кола користувачів. Істотною його перевагою є можливість використання раніше розроблених програм. Його недоліками є невисока швидкодія, функціональна і структурна надлишковість комп'ютерних засобів.

Достатньо ефективним шляхом покращення продуктивності нейрокомп'ютерних систем вважається перехід від програмних емуляторів до апаратних реалізацій, які передбачають декілька різних рівнів реалізації так званих нейрокомп'ютерів – від створення плат розширення для прискорення виконання базових операцій (другий варіант) і аж до розробки повністю спеціалізованої елементної бази, що забезпечує побудову пристроїв в нейромережному елементному базисі (третій варіант).

Другий варіант є перспективним, оскільки він передбачає поєднання універсальних і спеціальних засобів. Таке поєднання забезпечує високу ефективність використання обладнання при створенні систем для опрацювання у реальному часі потоків даних за алгоритмами, які є нерегулярними з великою кількістю логічних операцій. При цьому розробка нейрокомп'ютерних систем з заданими технічними параметрами зводиться до доповнення обчислювального ядра додатковими апаратно-програмними компонентами.

Третій варіант орієнтований на обробку у реальному часі інтенсивних потоків даних. При цьому висока ефективність використання обладнання досягається узгодженням обчислювальної здатності апаратних засобів з інтенсивністю надходження потоків даних. Використання для побудови нейрокомп'ютерних систем обчислювальних полів на основі програмованих логічних інтегральних схем (ПЛІС) з динамічним репрограмуванням відкриває новий етап створення реконфігурованих нейрокомп'ютерних систем (РНС). Особливістю РНС є архітектурна гнучкість, яка однаково стосується як апаратних, так і програмних засобів. В РНС структура може динамічно змінюватися як при підготовці до розв'язання задачі, так і в ході обчислювального процесу. Використання принципу реконфігурованості при побудові нейрокомп'ютерних систем забезпечує високу живучість і нарощування функцій.

Проаналізовано програмні, мікропрограмні та апаратні засоби реалізації нейромереж. Виявлено, що забезпечення режиму реального часу потребує великих витрат обладнання, вартості та термінів розроблення. Відомі програмні та

мікропрограмні засоби не враховують вимог конкретних застосувань щодо габаритів і споживаної потужності, а апаратні засоби мають низьку ефективність використання обладнання. Встановлено, що всі види реалізації нейроалгоритмів у безпосередньому вигляді зустрічаються доволі рідко. Для створення нейромереж реального часу доцільно використовувати комбіновані підходи з переважанням апаратних засобів, які реалізують розпаралелювання обчислення не тільки у часі, а й у просторі. Для таких обчислень характерне введення додаткового обладнання, відсутність проміжних пересилань інформації, а також апаратне управління.

#### 1.4. Постановка задачі магістерської роботи

Постановка завдання.

1. Провести аналіз існуючих рішень для побудови моделей та структур нейроелемента з паралельно-груповим обчисленням.
2. Розробити структурну схему апаратного пристрою нейроелемента з паралельно-груповим обчисленням.
3. Здійснити моделювання роботи нейроелемента з паралельно-груповим обчисленням.

## 2 АЛГОРИТМИ РЕАЛІЗАЦІЇ БАЗОВИХ НЕЙРООПЕРАЦІЙ

### 2.1 Вибір принципів та формування вимог до компонентів комп'ютерних штучних нейронних мереж реального часу .

Архітектури ШНМ реального часу повинні в повній мірі використовувати можливості НВІС-технології, враховувати вартість площі кристала, а також кількість вхідних і вихідних виводів. Число зовнішніх виводів НВІС обмежене рівнем технології та розміром кристалу. В основу побудови ШНМ реального часу пропонується покласти принципи, які дозволять зменшити вартість, терміни і розширити галузі їх застосування. Аналіз показує [10], що забезпечити дані вимоги можна при використанні таких принципів:

—модульності, який передбачає розробку компонентів АЗ у вигляді функціонально завершених пристроїв (модулів), що мають вихід на стандартний інтерфейс;

—узгодженості інтенсивності надходження даних з обчислювальною здатністю АЗ;

—конвеєризації та просторового паралелізму обробки даних;

—однорідності та регулярності архітектури ШНМ;

—локалізації та спрощення зв'язків між елементами АЗ;

—відкритості програмного забезпечення, що передбачає можливості нарощування та його вдосконалення, максимального використання стандартних драйверів та програмних засобів;

—спеціалізації та адаптації апаратно-програмних засобів до структури алгоритмів обробки та інтенсивності надходження даних;

—програмованості архітектури шляхом використання репрограмованих логічних інтегральних мікросхем.

### 2.2 Паралельно-вертикальні алгоритми та структури пристроїв групового підсумовування



У загальному випадку обчислення макрооперації групового підсумовування записується так:

$$Z = \sum_{j=1}^M C_j, \quad (2.2)$$

де  $M$  – кількість доданків;  $C_j$  –  $j$ -й доданок [1].

Нехай доданки  $C_j$  є двійковими  $n$ -розрядними додатними числами меншими за одиницю, які записуються так:

$$C_j = \sum_{i=1}^n 2^{-i} C_{ji}. \quad (2.3)$$

Підставивши значення у формулу , отримаємо:

$$Z = \sum_{j=1}^M \sum_{i=1}^n 2^{-i} C_{ji}. \quad (2.4)$$

Формула (2.3) відображає горизонтальну модель обчислення оператора групового підсумовування. У роботі [2] розглянуті всі можливі варіанти реалізації горизонтальної моделі групового підсумовування. Найшвидшим варіантом реалізації горизонтальної моделі групового підсумовування є паралельно-паралельний метод обчислення, граф алгоритму якого поданий на Рисунок. 2.

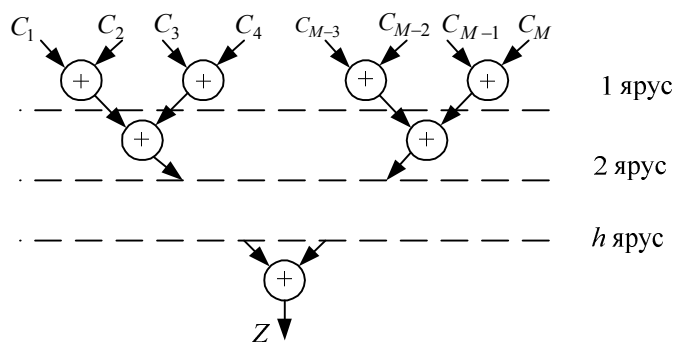


Рисунок 2.1 - Граф алгоритму паралельно-паралельного групового підсумовування.

Алгоритм паралельно-паралельного групового підсумовування є каскадним. Час обчислення суми макрооперації групового підсумовування за таким алгоритмом залежить від висоти графа (кількості ярусів), яка обчислюється так:

$$h = \lceil \log_2 M \rceil,$$

де  $\lceil \cdot \rceil$  – операція округлення до більшого цілого числа.

У кожному ярусі операнди розбиваються на пари, для кожної з яких обчислюється сума. Загальна кількість операцій додавання для обчислення суми макрооперації групового підсумовування рівна:

$$U = \frac{M}{2} + \frac{M}{4} + \frac{M}{8} + \dots + 1 = M - 1.$$

Підвищити швидкодію обчислення, ефективність використання та орієнтувати структуру макрооперації групового підсумовування на НВІС-реалізацію можна шляхом використання вертикального та багатооперандного додавання [3-7].

У неймережах для ефективного обчислення в реальному часі оператора групового підсумовування доцільно використовувати вертикальний та багатооперандний підходи для його реалізації. При використанні даних підходів

процес обчислення оператора групового підсумовування розглядається як виконання єдиної операції, що ґрунтується на базовій операції додавання значень бітів розрядного зрізу, тобто зводиться до вертикальної моделі обчислення. Замінивши у формулі (3) порядок підсумовування переходимо до вертикальної моделі обчислення оператора групового підсумовування, яка записується так:

$$Z = \sum_{i=1}^n 2^{-i} \sum_{j=1}^{M_i} C_{ji}, \quad (2.5)$$

де  $M_i$  – кількість доданків у  $i$ -у розрядному зрізі.

Існуючі вертикальні методи обчислення операції групового підсумовування зводять процес обчислення до перетворення багаторядного коду в однорядний. Таке перетворення ґрунтується на базовій операції перетворення трирядного коду в дворядний:

$$E = \begin{cases} C_{(j-1)1} \dots C_{(j-1)(n-1)} C_{(j-1)n} \\ + \\ C_{j1} \dots C_{j(n-1)} C_{jn} \\ + \\ C_{(j+1)1} \dots C_{(j+1)(n-1)} C_{(j+1)n} \end{cases} = \begin{cases} 0 S_1 \dots S_{n-1} S_n \\ + \\ P_0 P_1 \dots P_{n-1} 0 \end{cases}. \quad (2.6)$$

Перетворення трирядного коду в дворядний здійснюється за допомогою шару однорозрядних суматорів, що не мають зв'язків між собою. Для зменшення часу перетворення багаторядного коду в однорядний шари однорозрядних суматорів необхідно об'єднати за принципом дерева Уоллеса [3, 4]. Кількість шарів однорозрядних суматорів для обчислення оператора групового підсумовування визначається за формулою:

$$K = \lceil \log_{1.5} 0,5M \rceil. \quad (2.7)$$

Обчислення оператора групового підсумування за таким методом розглядається як виконання єдиної операції, де одиниці переносів враховуються тільки один раз при заключному етапі перетворення дворядного коду в однорядний.

Пришвидшення процесу перетворення багаторядного коду в однорядний пропонується здійснити шляхом використання для перетворення наступних однорозрядних операцій:

$$\begin{aligned}
 E_{3-2} = \begin{cases} C_{ji} \\ + \\ C_{(j+1)i} \\ + \\ C_{(j+2)i} \end{cases} &= \left\{ E_{7-3} = \begin{cases} C_{ji} \\ + \\ C_{(j+1)i} \\ + \\ C_{(j+2)i} \\ + \\ C_{(j+3)i} \\ + \\ C_{(j+4)i} \\ + \\ C_{(j+5)i} \\ + \\ C_{(j+6)i} \end{cases} = \begin{cases} P_{i-2} \\ + \\ S_{i-1} \\ + \\ S_i \end{cases} , \right. \\
 E_{15-4} = \begin{cases} C_{ji} \\ + \\ C_{(j+1)i} \\ + \\ C_{(j+2)i} \\ + \\ C_{(j+3)i} \\ + \\ C_{(j+4)i} \\ + \\ C_{(j+5)i} \\ + \\ C_{(j+6)i} \\ + \\ C_{(j+7)i} \\ + \\ C_{(j+8)i} \\ + \\ C_{(j+9)i} \\ + \\ C_{(j+10)i} \\ + \\ C_{(j+11)i} \\ + \\ C_{(j+12)i} \\ + \\ C_{(j+13)i} \\ + \\ C_{(j+14)i} \end{cases} &= \begin{cases} P \\ + \\ S \\ + \\ S \\ + \\ S \end{cases} \quad (2.8)
 \end{aligned}$$

де  $E_{3-2}$ ,  $E_{7-3}$  і  $E_{15-4}$  – результати однорозрядних операцій додавання відповідно трьох, семи і п'ятнадцяти операндів. Для реалізації таких операцій використовуються 3-, 7- і 15-входові однорозрядні суматори.

Аналітичні вирази для реалізації 7-входового однорозрядного суматора запишуться так:

$$\begin{aligned} S_2(2^0) &= Y_0 L_1 \vee Y_1 L_0 \vee Y_0 L_3 \vee Y_1 L_2 \vee Y_2 L_1 \vee Y_3 L_0 \vee Y_1 L_4 \vee Y_2 L_3 \vee Y_3 L_2 \vee Y_3 L_4, \\ S_1(2^1) &= Y_0 L_2 \vee Y_1 L_1 \vee Y_2 L_0 \vee Y_2 L_1 \vee Y_3 L_0 \vee Y_2 L_4 \vee Y_3 L_3 \vee Y_3 L_4 \vee Y_0 L_3 \vee Y_1 L_2, \\ P(2^2) &= Y_0 L_4 \vee Y_1 L_3 \vee Y_2 L_2 \vee Y_3 L_1 \vee Y_1 L_4 \vee Y_2 L_3 \vee Y_3 L_2 \vee Y_2 L_4 \vee Y_3 L_3 \vee Y_3 L_4, \end{aligned} \quad (2.9)$$

$$\begin{aligned} \text{де } Y_0 &= \overline{C_7 C_6 C_5}; \quad Y_1 = \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 C_5}; \quad Y_2 = \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 C_5}; \\ Y_3 &= C_7 C_6 C_5; \quad L_0 = \overline{C_4 C_3 C_2 C_1}; \quad L_1 = \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1}; \\ L_2 &= \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1}; \\ L_3 &= \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1}; \quad L_4 = C_4 C_3 C_2 C_1. \end{aligned}$$

На основі аналітичних виразів (2.9) синтезуємо комбінаційний 7-входовий однорозрядний суматор, схема якого подана на Рисунок. 2.2.

Швидкодія комбінаційного 7-входового однорозрядного суматора визначається часом затримки проходження даних з входу на вихід і обчислюється так:

$$t_{C_{M7-3}} = 5t_{\text{лог.І}},$$

де  $t_{\text{лог.І}}$  – час спрацювання логічного елемента І.

Розробку аналітичного виразу для синтезу 15-входового однорозрядного суматора будемо здійснювати поетапно.

На першому етапі розробки розбиваємо вхідні дані на чотири групи таким чином:  $L=C_1 C_2 C_3 C_4$ ,  $Y=C_5 C_6 C_7$ ,  $K=C_8 C_9 C_{10} C_{11}$ ,  $D=C_{12} C_{13} C_{14} C_{15}$ .

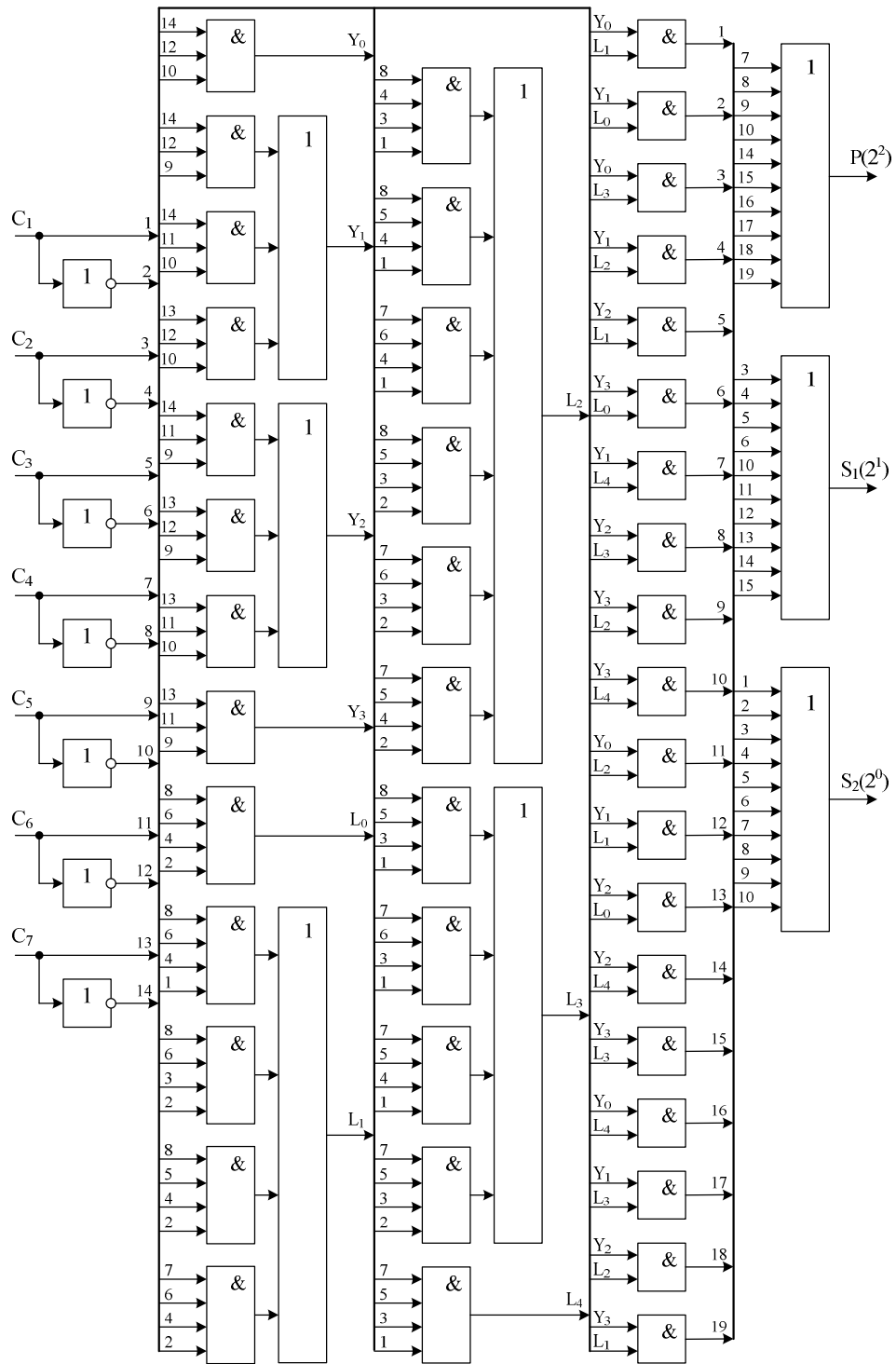


Рисунок 2.2 - Схема комбiнацiйного 7-входового однорозрядного суматора.

На другому етапi для кожної групи формуємо аналітичнi вирази для визначення кiлькостi одиниць у групi:

— група L

$$L_0 = \overline{C_4 C_3 C_2 C_1}; \quad L_1 = \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 \overline{C_1}} \vee \overline{C_4 C_3 \overline{C_2} C_1} \vee \overline{C_4 \overline{C_3} C_2 C_1};$$

$$L_2 = \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 \overline{C_2} C_1} \vee \overline{C_4 \overline{C_3} C_2 C_1} \vee \overline{C_4 C_3 C_2 \overline{C_1}} \vee \overline{C_4 \overline{C_3} C_2 \overline{C_1}} \vee \overline{C_4 C_3 \overline{C_2} \overline{C_1}};$$

$$L_3 = \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 \overline{C_3} C_2 C_1} \vee \overline{C_4 C_3 \overline{C_2} C_1} \vee \overline{C_4 C_3 C_2 \overline{C_1}}; \quad L_4 = \overline{C_4 C_3 C_2 C_1}.$$

— група Y

$$Y_0 = \overline{C_7 C_6 C_5}; \quad Y_1 = \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 \overline{C_5}} \vee \overline{C_7 \overline{C_6} C_5}; \quad Y_2 = \overline{C_7 C_6 C_5} \vee \overline{C_7 \overline{C_6} C_5} \vee \overline{C_7 C_6 \overline{C_5}};$$

$$Y_3 = \overline{C_7 C_6 C_5}.$$

— група K

$$K_0 = \overline{C_{11} C_{10} C_9 C_8}; \quad K_1 = \overline{C_{11} C_{10} C_9 C_8} \vee \overline{C_{11} C_{10} C_9 \overline{C_8}} \vee \overline{C_{11} C_{10} \overline{C_9} C_8} \vee \overline{C_{11} \overline{C_{10}} C_9 C_8};$$

$$K_2 = \overline{C_{11} C_{10} C_9 C_8} \vee \overline{C_{11} C_{10} \overline{C_9} C_8} \vee \overline{C_{11} \overline{C_{10}} C_9 C_8} \vee \overline{C_{11} C_{10} C_9 \overline{C_8}} \vee \overline{C_{11} \overline{C_{10}} C_9 \overline{C_8}} \vee \overline{C_{11} C_{10} \overline{C_9} \overline{C_8}};$$

$$K_3 = \overline{C_{11} C_{10} C_9 C_8} \vee \overline{C_{11} \overline{C_{10}} C_9 C_8} \vee \overline{C_{11} C_{10} \overline{C_9} C_8} \vee \overline{C_{11} C_{10} C_9 \overline{C_8}}; \quad K_4 = \overline{C_{11} C_{10} C_9 C_8}.$$

— група D

$$D_0 = \overline{C_{15} C_{14} C_{13} C_{12}}; \quad D_1 = \overline{C_{15} C_{14} C_{13} C_{12}} \vee \overline{C_{15} C_{14} C_{13} \overline{C_{12}}} \vee \overline{C_{15} C_{14} \overline{C_{13}} C_{12}} \vee \overline{C_{15} \overline{C_{14}} C_{13} C_{12}};$$

$$D_2 = \overline{C_{15} C_{14} C_{13} C_{12}} \vee \overline{C_{15} C_{14} \overline{C_{13}} C_{12}} \vee \overline{C_{15} \overline{C_{14}} C_{13} C_{12}} \vee \overline{C_{15} C_{14} C_{13} \overline{C_{12}}} \vee \overline{C_{15} \overline{C_{14}} C_{13} \overline{C_{12}}} \vee \overline{C_{15} C_{14} \overline{C_{13}} \overline{C_{12}}};$$

$$D_3 = \overline{C_{15} C_{14} C_{13} C_{12}} \vee \overline{C_{15} \overline{C_{14}} C_{13} C_{12}} \vee \overline{C_{15} C_{14} \overline{C_{13}} C_{12}} \vee \overline{C_{15} C_{14} C_{13} \overline{C_{12}}}; \quad D_4 = \overline{C_{15} C_{14} C_{13} C_{12}}.$$

На третьому етапі записуємо аналітичні вирази для груп R і H, які об'єднують відповідно групи L і Y та групи K і D:

$$R_0 = Y_0 L_0; \quad R_1 = Y_1 L_0 \vee Y_0 L_1; \quad R_2 = Y_2 L_0 \vee Y_1 L_1 \vee Y_0 L_2; \quad R_3 = Y_3 L_0 \vee Y_2 L_1 \vee Y_1 L_2 \vee Y_0 L_3;$$

$$R_4 = Y_3 L_1 \vee Y_2 L_2 \vee Y_1 L_3 \vee Y_0 L_4; \quad R_5 = Y_3 L_2 \vee Y_2 L_3 \vee Y_1 L_4; \quad R_6 = Y_3 L_3 \vee Y_2 L_4; \quad R_7 = Y_3 L_4;$$

$$H_0 = K_0 D_0; \quad H_1 = K_1 D_0 \vee K_0 D_1; \quad H_2 = K_2 D_0 \vee K_1 D_1 \vee K_0 D_2;$$

$$H_3 = K_3 D_0 \vee K_2 D_1 \vee K_1 D_2 \vee K_0 D_3; \quad H_4 = K_4 D_0 \vee K_3 D_1 \vee K_2 D_2 \vee K_1 D_3 \vee K_0 D_4;$$

$$H_5 = K_4 D_1 \vee K_3 D_2 \vee K_2 D_3 \vee K_1 D_4; \quad H_6 = K_4 D_2 \vee K_3 D_3 \vee K_2 D_4;$$

$$H_7 = K_4 D_3 \vee K_3 D_4; \quad H_8 = K_4 D_4.$$

На четвертому етапі на основі попередніх аналітичних виразів запишемо аналітичні вирази визначення кількості одиниць у вхідних даних, що надходять зі всіх 15 входів:

$$F_1 = R_0 H_1 \vee R_1 H_0; \quad F_2 = R_2 H_0 \vee R_1 H_1 \vee R_0 H_2; \quad F_3 = R_3 H_0 \vee R_2 H_1 \vee R_1 H_2 \vee R_0 H_3;$$

$$F_4 = R_4 H_0 \vee R_3 H_1 \vee R_2 H_2 \vee R_1 H_3 \vee R_0 H_4; \quad F_5 = R_5 H_0 \vee R_4 H_1 \vee R_3 H_2 \vee R_2 H_3 \vee R_1 H_4 \vee R_0 H_5;$$

$$F_6 = R_6 H_0 \vee R_5 H_1 \vee R_4 H_2 \vee R_3 H_3 \vee R_2 H_4 \vee R_1 H_5 \vee R_0 H_6;$$

$$F_7 = R_7 H_0 \vee R_6 H_1 \vee R_5 H_2 \vee R_4 H_3 \vee R_3 H_4 \vee R_2 H_5 \vee R_1 H_6 \vee R_0 H_7;$$

$$F_8 = R_7 H_1 \vee R_6 H_2 \vee R_5 H_3 \vee R_4 H_4 \vee R_3 H_5 \vee R_2 H_6 \vee R_1 H_7 \vee R_0 H_8;$$

$$F_9 = R_7 H_2 \vee R_6 H_3 \vee R_5 H_4 \vee R_4 H_5 \vee R_3 H_6 \vee R_2 H_7 \vee R_1 H_8;$$

$$F_{10} = R_7 H_3 \vee R_6 H_4 \vee R_5 H_5 \vee R_4 H_6 \vee R_3 H_7 \vee R_2 H_8; \quad F_{11} = R_7 H_4 \vee R_6 H_5 \vee R_5 H_6 \vee R_4 H_7 \vee R_3 H_8;$$

$$F_{12} = R_7 H_5 \vee R_6 H_6 \vee R_5 H_7 \vee R_4 H_8; \quad F_{13} = R_7 H_6 \vee R_6 H_7 \vee R_5 H_8; \quad F_{14} = R_7 H_7 \vee R_6 H_8; \quad F_{15} = R_7 H_8.$$

На п'ятому етапі на основі попередніх аналітичних виразів запишемо аналітичні вирази для синтезу 15-входового однорозрядного суматора:

$$S_3(2^0) = F_1 \vee F_3 \vee F_5 \vee F_7 \vee F_9 \vee F_{11} \vee F_{13} \vee F_{15},$$

$$S_2(2^1) = F_2 \vee F_3 \vee F_6 \vee F_7 \vee F_{10} \vee F_{11} \vee F_{14} \vee F_{15},$$

$$S_1(2^2) = F_4 \vee F_5 \vee F_6 \vee F_7 \vee F_{12} \vee F_{13} \vee F_{14} \vee F_{15},$$

$$P(2^3) = F_8 \vee F_9 \vee F_{10} \vee F_{11} \vee F_{12} \vee F_{13} \vee F_{14} \vee F_{15}.$$

Аналогічно можна розробляти аналітичні вирази для синтезу багатовходових однорозрядних суматорів з більшою кількістю входів.

Для групового підсумовування багаторозрядних чисел використовуються багатовходові однорозрядні суматори, які синтезуються за вище розробленими аналітичними виразами. Об'єднання таких суматорів за принципом дерева Уоллеса забезпечує перетворення багаторядного коду в дворядний, який перетворюється в однорядний за допомогою паралельного суматора. На Рисунок. 2.3 подано приклад схеми пристрою для підсумовування восьми чисел розрядністю вісім.

Підсумовування чисел у даному пристрої здійснюється за чотири етапи шляхом використання таких перетворень: 7-рядного коду в 3-рядний, 3-рядного коду в 2-рядний та 2-рядного коду в 1-рядний код.

На першому етапі обчислення за допомогою шару з восьми 7-входових однорозрядних суматорів, які не зв'язані між собою, здійснюється перетворення 7-рядного коду в 3-рядний. На другому та третьому етапах за допомогою двох шарів, кожний з яких складається з дев'яти 3-входових однорозрядних суматорів, які не зв'язані між собою, здійснюється перетворення з 3-рядного коду в 2-рядний. На завершальному четвертому етапі за допомогою 9-розрядного паралельного суматора виконується перетворення 2-рядного коду в 1-рядний.



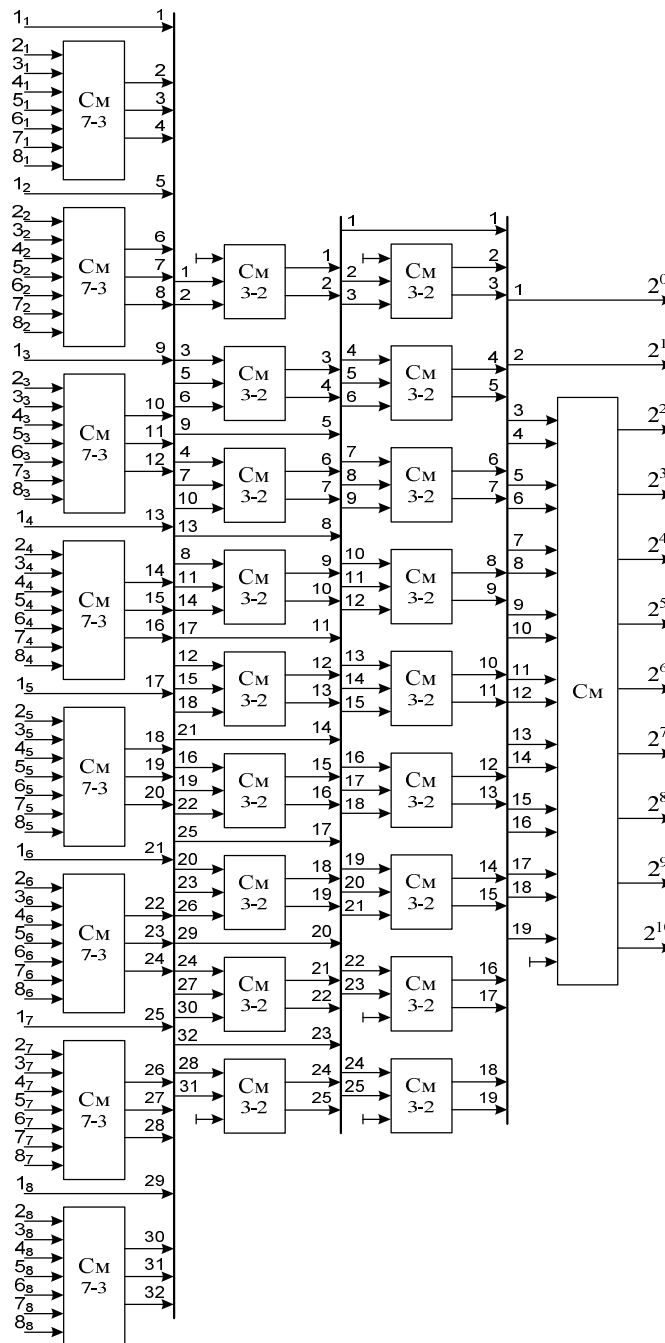


Рисунок 2.3 - Схема пристрою для підсумовування восьми чисел розрядністю вісім.

Час підсумовування в даному пристрої визначається за такою формулою:

$$t_{BCM} = t_{CM7-3} + 2t_{CM3-2} + t_{CM2-1},$$

де  $t_{CM7-3}$  – час перетворення 7-рядного коду в 3-рядний,  $t_{CM3-2}$  – час перетворення 3-рядного коду в 2-рядний,  $t_{CM2-1}$  – час перетворення 2-рядного коду в 1-рядний.

Розроблені 7-входовий однорозрядний суматор та пристрій для підсумовування восьми чисел розрядністю вісім були описані на мові описання апаратних засобів VHDL у середовищі Xilinx ISE 13.1, відмодельовані вбудованими у дане середовище засобами та реалізовані на основі ПЛІС Xilinx Spartan3E xc3s500e-5fg320. Ці пристрої можуть бути використані як компоненти для синтезу нейроелементів.

### 2.3. Паралельно-вертикальні алгоритми та НВІС-структури пристроїв обчислення суми квадратів різниць

Метод паралельно-вертикального обчислення суми квадратів різниць. Для обчислення суми різниць квадратів використаємо багатооперандний вертикальний підхід, який передбачає одночасне послідовно-порозрядне надходження операндів, формування та підсумовування в кожному такті макрочасткових результатів. Вертикальний метод обчислення суми квадратів різниць вимагає, щоб операнди були представлені в порозрядному вигляді згідно з формулою [7,9,10]:

$$X = \sum_{j=1}^n 2^{-j} x_j ,$$

де  $x_j$  – значення  $j$ -го розряду операнда; –  $n$  розрядність операндів.

Паралельно-вертикальне обчислення суми квадратів різниць ґрунтується на алгоритмі вертикального піднесення до квадрату:

$$X^2 = (0.01) \wedge x_1 + 2^{-1}(0.x_1 01) \wedge x_2 + 2^{-2}(0.x_1 x_2 01) \wedge x_3 + \dots + 2^{-(n-1)}(0.x_1 x_2 \dots x_{n-1} 01) \wedge x_n = \sum_{j=1}^n 2^{-(j-1)} R_j \quad (2.10)$$

де  $R_j$  – частковий результат піднесення до квадрату, який визначається так:

$$R_j = (0.x_1 x_2 \dots x_{j-1} 01) \wedge x_j. \quad (2.11)$$

Використовуючи багатооперандний підхід і алгоритм вертикального піднесення до квадрату операцію обчислення суми квадратів різниць можна звести до формування і підсумовування макрочасткових результатів відповідно до формули:

$$y = (X_1^e - X_1^b)^2 + (X_2^e - X_2^b)^2 + \dots + (X_N^e - X_N^b)^2 = \Delta X_1^2 + \Delta X_2^2 + \dots + \Delta X_N^2 = \sum_{k=1}^m 2^{-(k-1)} P_k, \quad (2.12)$$

$$P_k = \sum_{i=1}^N R_{ki}, \quad (2.13)$$

де  $m$  – розрядність різниць  $\Delta X_i$ ;  $R_{ki}$  –  $k$ -ий частковий результат піднесення до квадрату  $i$ -ої різниці  $\Delta X_i$ .

Підставляючи значення (2.3.4) у формулу (2.3.3) зведемо обчислення суми квадратів різниць до операції багатооперандного підсумовування:

$$y = \sum_{i=1}^N \Delta X_i^2 = \sum_{k=1}^m \sum_{i=1}^N 2^{-(k-1)} R_{ki}. \quad (2.14)$$

Основними етапами паралельно-вертикального методу обчислення суми квадратів різниць є:

— одночасне послідовно-порозрядне надходження операндів  $X_i^e, X_i^b$  і обчислення модуля  $\Delta X_i$ ;

- формування часткових результатів піднесення до квадрату  $R_{ki}$ ;
- формування макрочасткового результату обчислення  $P_k$  шляхом підсумовування часткових результатів піднесення до квадрату  $R_{ki}$ ;
- отримання результату суми квадратів різниць шляхом підсумовування із зсувом вправо на один розряд макрочасткових результатів обчислення  $P_k$ .

Структура пристрою для паралельно-вертикального обчислення суми квадратів різниць. Структура пристрою паралельно-вертикального обчислення суми квадратів різниць повинна бути орієнтовані на НВІС-реалізацію, враховувати вартість площі кристала, а також кількість вхідних і вихідних виводів [11,14]. Число зовнішніх виводів НВІС-пристроїв обмежене рівнем технології та розміром кристалу. В основу побудови НВІС-пристроїв реального часу для паралельно-вертикального обчислення суми квадратів різниць пропонується покласти такі принципи [14,15]:

- узгодженості інтенсивності надходження даних з обчислювальною інтенсивністю пристрою;
- конвеєризації та просторового паралелізму;
- регулярності, модульності та широкого використання стандартних елементів;
- локалізації та зменшення кількості зв'язків між елементами пристрою.

В залежності від способу формування та підсумовування макрочасткових результатів  $P_k$  можливі такі варіанти реалізації пристрою обчислення суми квадратів різниць:

- з послідовним формуванням і підсумовуванням макрочасткових результатів  $P_k$ ;
- з паралельним формуванням і послідовним підсумовуванням макрочасткових результатів  $P_k$ ;
- з паралельним формуванням і підсумовуванням макрочасткових результатів  $P_k$ .

При паралельно-вертикальному обчисленні суми квадратів різниць використовується паралельне формування макрочасткового результату  $P_k$  з

послідовним його підсумовуванням. Структура пристрою з паралельно-вертикальним обчисленням суми квадратів різниць наведена на Рисунок.2, де Від – віднімач, Тг – тригер, Рг – регістр, ПК – перетворювач кодів, БСм – багатовходовий суматор, См – суматор, Бл – блок формування часткових результатів піднесення до квадрату.

У даному пристрої надходження чисел  $X_i^e$  і  $X_i^b$  здійснюється одночасно послідовним кодом молодшими розрядами вперед. В кожному блоці Бл і за допомогою віднімача Від обчислюється різниця  $\Delta X_i$ , яка записується в регістр Рг  $\Delta X_i$ . Обчислена різниця  $\Delta X_i$  надходить на входи перетворювача кодів ПК, на виході якого отримуємо її модуль  $|\Delta X_i|$ . У наступних тактах роботи в кожному блоці Бл і формувачем  $R_k$  формується частковий результат піднесення до квадрату  $R_k$ . Формування часткових результатів піднесення до квадрату  $R_k$  здійснюється порозрядно починаючи з молодших розрядів модуля  $|\Delta X_i|$  у відповідності до формули (2). Сформовані в блоках Бл і часткові результати піднесення до квадрату  $R_{ki}$  надходять на входи багатовходового суматора БСм, де виконується їх підсумовування. Отримана сума є макрочастковим результатом обчислення суми квадратів різниць і вона записується в регістр Рг  $R_k$ . На суматорі См у відповідності з формулою (3) виконується додавання отриманого макрочасткового результату  $R_k$  до зсунутої вправо на один розряд накопиченої суми макрочасткових результатів.

Обчислення модуля  $|\Delta X_i|$ , формування і підсумовування макрочасткових результатів  $R_k$  елементів вибірок суміщається в часі з обчисленням  $\Delta X_i$  наступних елементів вибірок. При такому суміщенні для забезпечення режиму реального часу дані операції повинні виконуватися за час:

$$t_{об} = nT_d,$$

де  $n$  – розрядність операндів;  $T_d$  – період надходження даних.

Для обчислення суми квадратів різниць в реальному часі в даному пристрої використовується конвеєр, який повинен працювати з тактом:

$$T_k \leq \frac{t_{Об}}{m+1}.$$

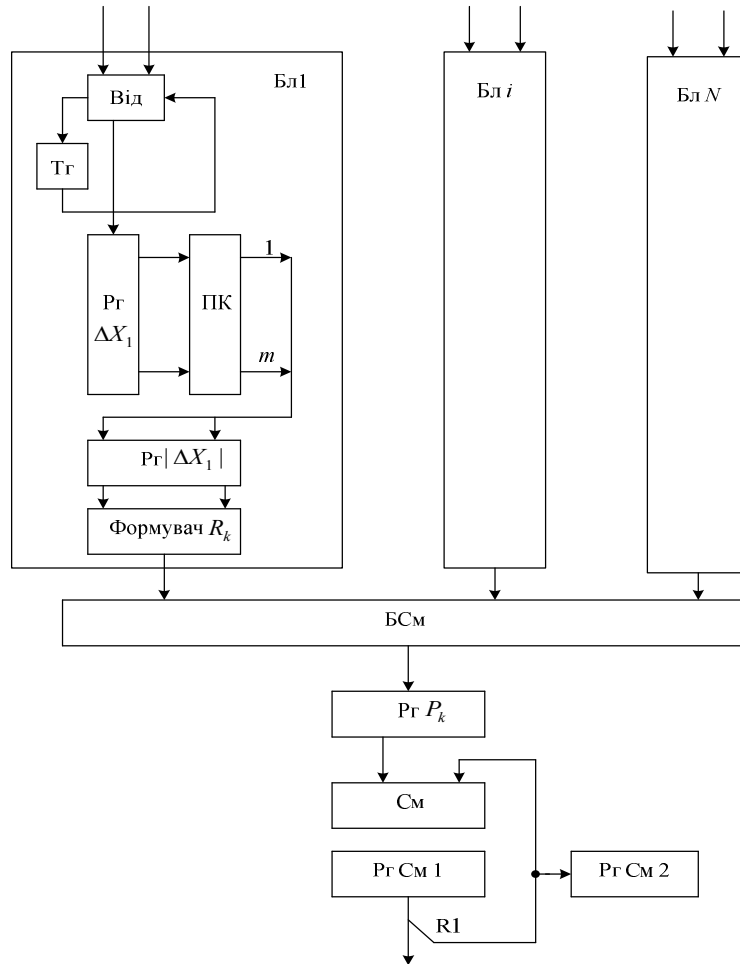


Рисунок 2.4 - Структура пристрою паралельно-вертикального обчислення суми квадратів різниць.

## 2.4 Паралельно-вертикальні алгоритми та структури пристроїв обчислення максимальних і мінімальних значень

На сучасному етапі розвитку інформаційних технологій намітилась тенденція збільшення ролі логічних обчислень. Такі обчислення є домінуючими при реалізації алгоритмів Data Mining. Логічні обчислення мають ряд специфічних особливостей, які не дозволяють при їх реалізації використовувати

відомі обчислювальні методи і алгоритми. Існуючі апаратні засоби в основному орієнтовані на реалізацію алгоритмів з перевагою обчислювальних операцій над логічними і вони не враховують специфіки обробки логічних даних. Особливістю реалізації логічних обчислень на сучасних апаратних засобах є неефективність використання багаторозрядних операційних пристроїв, що істотно знижує продуктивність та ефективність використання обладнання. У зв'язку з цим особливої актуальності набуває проблема розробки нових ефективних методів і алгоритмів логічних обчислень, орієнтованих на НВІС-реалізацію [1,2,3]. Для переходу від алгоритмів логічних обчислень до НВІС-архітектур доцільно використовувати методи просторово-часового відображення таких алгоритмів у паралельні однорідні структури з високою ефективністю використання обладнання. Процес відображення алгоритмів у паралельні спеціалізовані НВІС-архітектури є складним і вимагає взаємної адаптації як алгоритмів, так і структур обчислювальних засобів.

Аналіз методів і алгоритмів обчислення максимальних і мінімальних значень показав, що для НВІС-реалізацій найефективнішими алгоритмами є ті, що ґрунтуються на методі порозрядного порівняння [4]. Обчислення максимального  $A_{\max}$  і мінімального  $A_{\min}$  чисел з групи чисел  $A_1, A_2, \dots, A_j, \dots, A_m$  за таким алгоритмом виконується послідовним порівнянням розрядів всіх чисел починаючи зі старшого. При кожному порівнянні отримуємо  $i$ -і розряди максимального і мінімального чисел, обчислення яких здійснюється за формулами:

$$\overline{A_{i \max}} = \bigwedge_{j=1}^m \overline{a_{ji} \wedge y_{ij}}, \quad y_{1j} = 1; \quad (2.15)$$

$$A_{i \min} = \bigwedge_{j=1}^m \overline{a_{ji} \wedge z_{ij}}, \quad z_{1j} = 1 \quad (2.16)$$

де  $y_{ij}, z_{ij}$  -  $i$ -і розряди  $j$ -х слів управління;  $a_{ji}$  -  $i$ -й розряд  $j$ -о числа;  $m$  - кількість чисел у групі.

Формування  $(i+1)$ -х розрядів  $j$ -х слів управління виконується за формулами:

$$y_{(i+1)j} = (\overline{A_{i\max}} \vee x_{ji}) \wedge y_{ij}, \quad (2.17)$$

$$z_{(i+1)j} = (A_{i\min} \vee x_{ji}) \wedge z_{ij}. \quad (2.18)$$

Процес синтезу паралельно-вертикальних НВІС-структур для обчислення максимальних і мінімальних значень зводиться до виконання наступних етапів:

- виділення базової операції та використання її для просторово-часового відображення алгоритму;
- розробка схеми процесорного елемента (ПЕ), що реалізує базову операцію алгоритму;
- синтез НВІС-структур на базі ПЕ;
- організація інтерфейсу НВІС.

Аналіз алгоритму паралельно-вертикального обчислення максимальних і мінімальних значень дозволив виділити для НВІС-реалізації базову операцію. Така базова операція включає формування розрядів слів управління за формулами (5.3) і (5.4) та виконання наступних логічних обчислень:

$$\overline{A_{ji\max}} = \overline{a_{ji} \wedge y_{ij}}, \quad (2.19)$$

$$A_{ji\min} = \overline{a_{ji} \wedge z_{ij}}. \quad (2.20)$$

Для реалізації базової операції алгоритму паралельно-вертикального обчислення максимальних і мінімальних значень (формули (2.19) – (2.20)) розроблено схему ПЕ та на його базі синтезовано пристрій (Рисунок.2.5). Особливістю даного пристрою є спільні шини результатів, підключення до яких здійснюється за допомогою логічних елементів 2І-НЕ з відкритим колектором.

Для збільшення ефективності визначення максимального числа з групи чисел доцільно застосувати паралельно-вертикальний підхід, при якому час визначення максимального числа з групи чисел не залежав від кількості чисел.



При цьому у кожному такті роботи на інформаційні входи пристрою надходять розрядні зрізи всіх чисел починаючи зі старших розрядів.

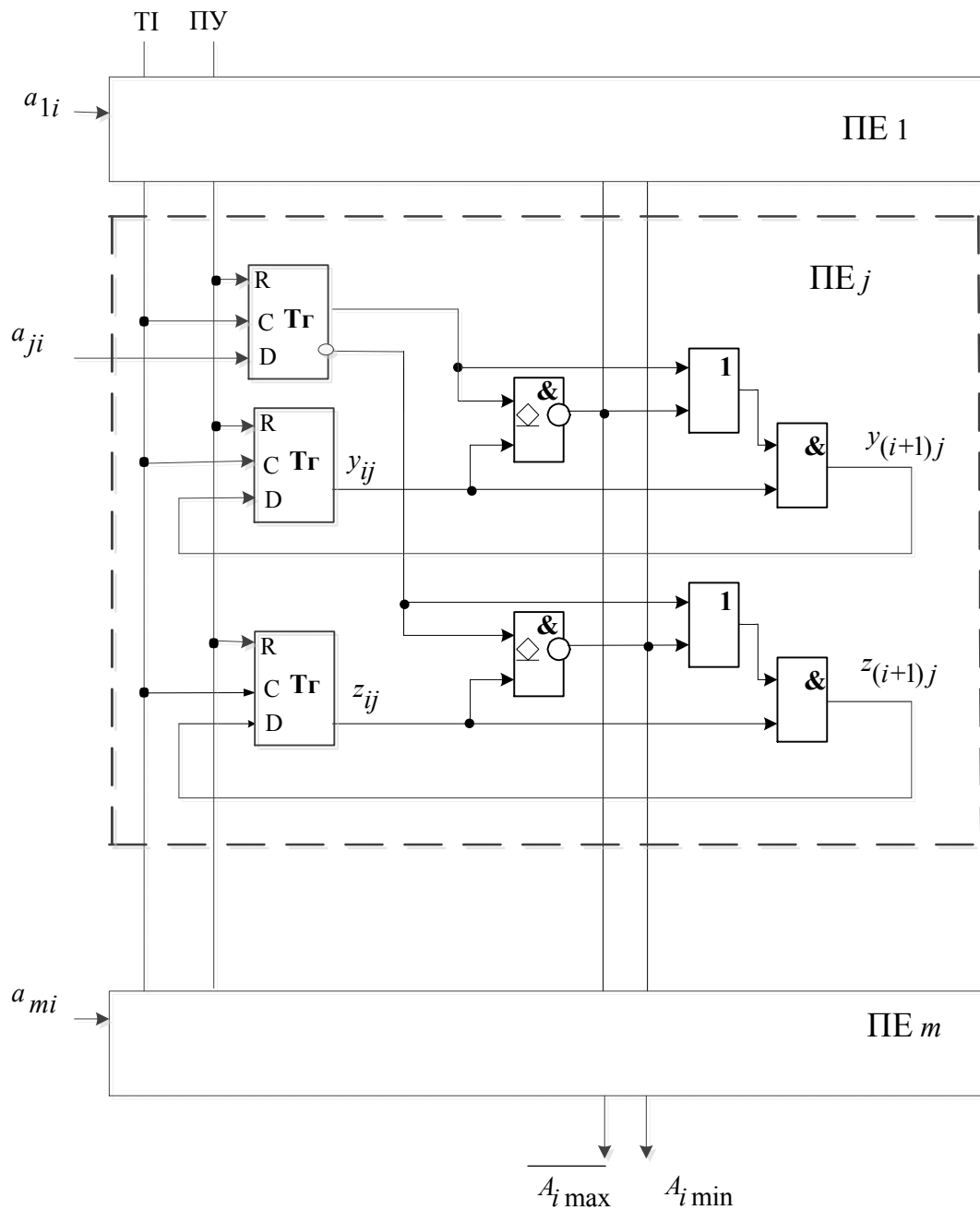


Рисунок 2.5 - Схема пристрою паралельно-вертикального обчислення максимальних і мінімальних значень.

Структуру такого пристрою наведено на рисунку. Рисунок 2.5, де TI – тактовий вхід, У – вхід початкової установки тригерів,  $x_1, \dots, x_m$  – однорозрядні

інформаційні входи, де  $m$  – кількість чисел, що порівнюються,  $БП_1, \dots, БП_m$  – блоки порівняння,  $T_1$  і  $T_2$  – D-тригери, Вих – вихід результату.

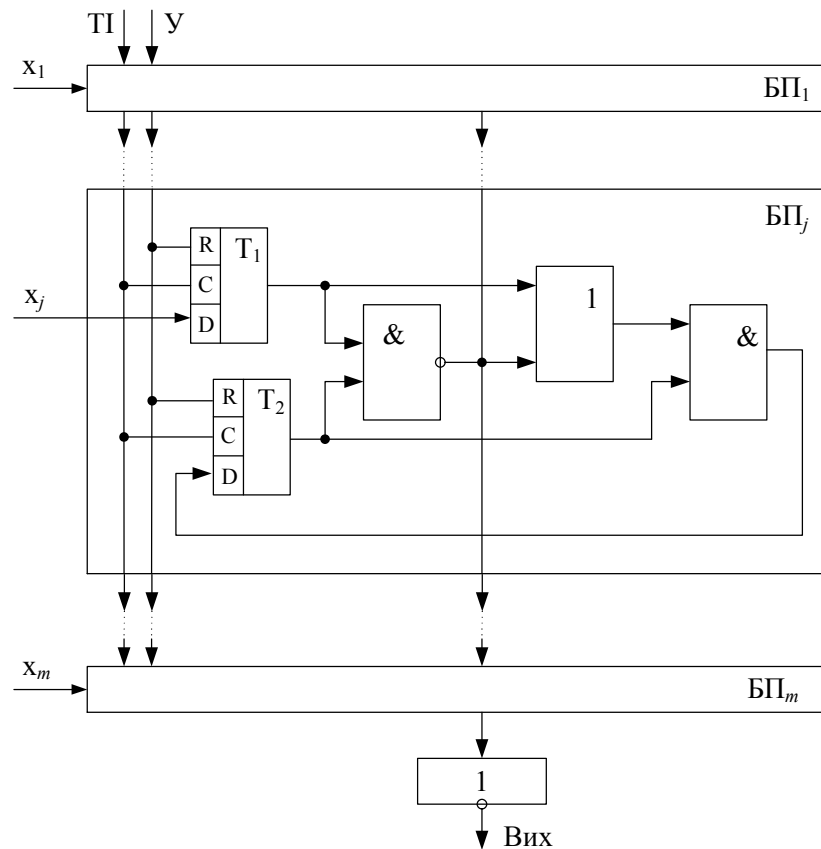


Рисунок 2.6 - Структура пристрою визначення максимального числа з групи чисел.

Пристрій для визначення максимального числа з групи чисел працює наступним чином.

Перед початком роботи імпульсом початкової установки, який надходить із входу початкової установки  $У$ , тригери  $T_1$  та  $T_2$  у кожному блоці порівняння  $БП_j$  ( $j=1, \dots, m$ ) встановлюються у лог.1. Інформація з виходів тригерів  $T_1$  і  $T_2$  (лог.1) у кожному блоці порівняння  $БП_j$  встановлює на виходах елементів І сигнал лог.1.

У першому такті у кожному блоці порівняння  $БП_j$  на інформаційний вхід тригера  $T_2$  з елемента І надходить значення лог.1, а на інформаційний вхід тригера  $T_1$  з однорозрядного інформаційного входу  $x_j$  надходить значення

старшого розряду  $j$ -о числа. Першим тактовим імпульсом у кожному блоці порівняння  $БП_j$  у тригер  $T_1$  записується старший розряд  $j$ -о числа, а у тригер  $T_2$  записується лог.1, яка дозволяє участь  $j$ -о числа у визначенні максимального значення. При цьому в кожному блоці порівняння  $БП_j$  значення старшого розряду  $j$ -о числа з виходу тригера  $T_1$  поступає на перший вхід елемента І-НЕ з відкритим колектором та перший вхід елемента АБО. Лог.1 з виходу тригера  $T_2$  (сигнал управління) надходить на другий вхід елемента І-НЕ з відкритим колектором та другий вхід елемента І. Інформація з виходів елементів І-НЕ з відкритим колектором блоків порівняння  $БП_1, \dots, БП_m$  об'єднується по монтажному І та поступає на другий вхід елементів АБО цих блоків порівняння і на вхід елемента НЕ. У випадку, коли старші розряди чисел, що порівнюються, рівні нулю, то на вході елемента НЕ формується лог.1, а у інших випадках – лог.0. Інформація з виходу елемента НЕ (старший розряд максимального числа) надходить на вихід результату Вих. При лог.1 на других входах елементів АБО блоків порівняння  $БП_1, \dots, БП_m$  на їхніх виходах встановлюється сигнал лог.1, а при лог.0 – інформація з виходів тригерів  $T_1$ . Інформація з виходів елементів АБО на виходах елементів І формує сигнали управління, які надходять на інформаційні входи  $T_2$ .

Другим тактовим імпульсом інформація з однорозрядного інформаційного входу  $x_j$  (наступний розряд) записується у тригер  $T_1$  кожного блоку порівняння  $БП_j$ , а у тригер  $T_2$  записується значення з виходу елемента І, яке дозволяє (лог.1) або забороняє (лог.0) участь інформації з однорозрядного інформаційного входу  $x_j$  у подальшому формуванні максимального числа.

Формування другого і наступних розрядів результату та сигналів управління виконуються так само, як у першому такті.

Час визначення максимального числа з групи чисел у цьому пристрої залежить від розрядності чисел, і не залежить від їхньої кількості. За  $n$  тактів, де  $n$  – розрядність чисел, отримаємо максимальне число з групи  $m$  чисел.

Час обчислення максимального числа у даному пристрої рівний:

$$t_{обч} = n(t_{T_2} + t_I),$$

де  $t_{T_2}$  – час запису інформації у тригер,  $t_I$  – час затримки інформації при проходженні через логічні елементи типу АБО, І, І-НЕ.

Затрати обладнання на реалізацію даного пристрою рівні:

$$W_{Пр} = n(2W_{T_2} + 3W_I)$$

де  $W_{T_2}$  – затрати обладнання на реалізацію тригера,  $W_I$  – затрати обладнання на реалізацію логічних елементів типу АБО, І, І-НЕ.

### 3 ВЕРТИКАЛЬНО-ПАРАЛЕЛЬНІ МЕТОДИ ОБЧИСЛЕННЯ ТА СИНТЕЗ АПАРАТНИХ КОМПОНЕНТІВ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

#### 3.1 Методи та структури апаратних компонентів для реалізації скалярного добутку

Традиційно обчислення скалярного добутку здійснюється за такою формулою:

$$Z = \sum_{j=1}^m A_j B_j = A_1 B_1 + A_2 B_2 + \dots + A_m B_m. \quad (0.1)$$

При обчисленні скалярного добутку за формулою (0.1) потрібно виконати  $m$  множень і  $m-1$  додавань [3].

Для паралельного обчислення скалярного добутку використаємо багатооперандний паралельно-вертикальний підхід, який передбачає паралельно-порозрядне надходження операндів, формування і підсумовування макрочасткових добутків [7-9].

Для реалізації даного методу обчислення оператора скалярного добутку необхідно, щоб множники  $A_j$  і  $B_j$  були представлені у порозрядному вигляді згідно з формулою [5]:

$$A_j = \sum_{i=0}^{n-1} (-1)^{2^i} 2^{-i} A_{ji}, \quad B_j = \sum_{i=0}^{n-1} (-1)^{2^i} 2^{-i} B_{ji}, \quad (0.2)$$

де  $A_{ji}, B_{ji}$  – цифри  $i$ -их розрядів множників  $A_j$  і  $B_j$ .

За формуванням макрочасткових добутоків методи обчислення скалярного добутку можна розділити на дві групи – з паралельним і з послідовним формуванням макрочасткових добутоків.

Формування макрочасткових добутоків можна здійснювати, починаючи з аналізу молодших чи старших розрядів множників  $B_j$ .

Обчислення скалярного добутку, починаючи з молодших розрядів множників  $B_j$ , здійснюється за два етапи: на першому етапі виконується накопичення множених  $A_j$ , а на другому етапі в кожному  $i$ -му такті ( $i=1, \dots, n$ ) виконується формування та накопичення макрочасткових добутоків згідно з формулами:

$$P_i = \sum_{j=1}^m A_j B_{ji}, \quad (0.3)$$

$$Z_i = 2^{-1} Z_{i-1} + P_i, \quad (0.4)$$

де  $P_i$  –  $i$ -й макрочастковий добуток;  $B_{ji}$  –  $i$ -й розряд  $j$ -о множника  $B_j$ ;  $Z_0=0$ .

Макрочастковий добуток  $P_i$  формується паралельним додаванням  $m$  часткових добутоків  $P_{ji} = A_j B_{ji}$  на  $m$ -входовому суматорі. Часткові добутки для кожної пари добутоків формуються згідно з формулою:

$$P_{ji} = \begin{cases} A_j, & \text{якщо } B_{ji} = 1 \\ 0, & \text{якщо } B_{ji} = 0 \end{cases}. \quad (0.5)$$

Структура пристрою обчислення скалярного добутку, який реалізує даний метод обчислення, наведена на рис. 0.1, де  $D1, \dots, Dm$  – інформаційні входи.

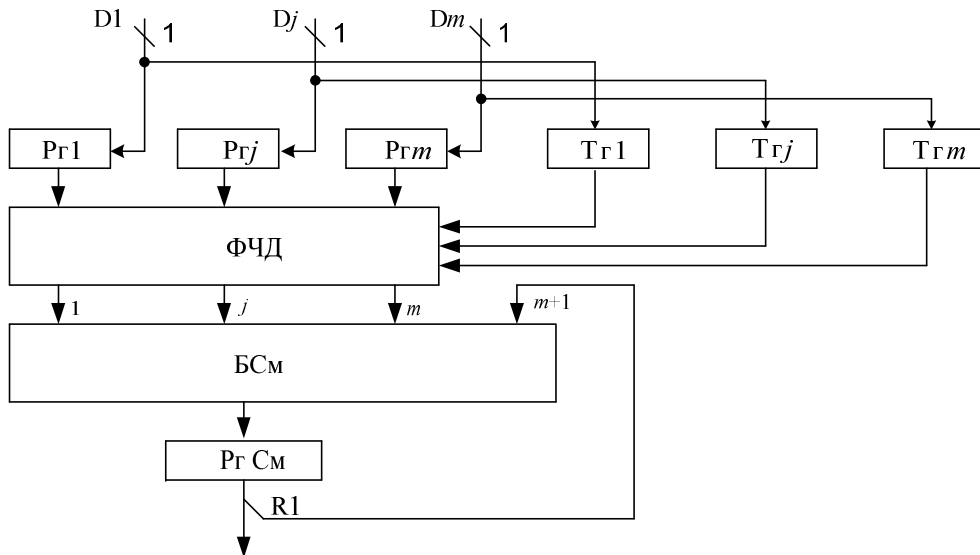


Рисунок 0.1. Структура пристрою обчислення скалярного добутку починаючи з молодших розрядів множників.

Основними компонентами такого пристрою є: тригери  $T_r$ , регістри  $P_r$ , формувач часткових добутків  $\Phi ЧД$  і багатовходовий суматор  $БСм$ . У кожному такті роботи такого пристрою здійснюється формування часткових добутків для всіх множників і їх додавання із попередньою сумою, зсунутою на один розряд вправо. У даному пристрої формування часткових добутків виконується за алгоритмом з аналізом одного розряду множників.

Затрати обладнання на реалізацію даного пристрою визначаються за таким виразом:

$$W_{\text{Пр1}} = (m + 1)W_{P_2} + mW_{T_2} + W_{\Phi ЧД1} + W_{(m+1)\text{-}См},$$

де  $W_{P_2}$ ,  $W_T$ ,  $W_{\Phi ЧД1}$ ,  $W_{(m+1)\text{-}См}$  – затрати обладнання відповідно на регістр, тригер, формувач часткових добутків,  $(m+1)$ -вхідний суматор.

Обчислення скалярного добутку в даному пристрої виконується за  $2n$  такти ( $n$  – розрядність операндів), кожний з яких рівний:

$$t_1 = t_{P_2} + t_{\Phi_{ЧД1}} + t_{(m+1)-С},$$

де  $t_{P_2}$ ,  $t_{\Phi_{ЧД1}}$ ,  $t_{(m+1)-С}$  – час спрацювання відповідно регістра, формувача часткових добутоків і  $(m+1)$ -вхідного суматора.

Використання при паралельно-вертикальному обчисленні скалярного добутку алгоритмів формування часткових добутоків з аналізом двох і більше розрядів множників  $B_j$  не забезпечує підвищення швидкодії, а тільки збільшує витрати обладнання. Підвищення швидкодії при обробці неперервних потоків даних можна досягнути при суміщенні процесів формування та накопичення макрочасткових добутоків для  $k$ -го масиву з введенням множених  $A_j$  для  $(k+1)$ -го масиву даних [10-12].

Зменшити кількість часткових добутоків при обчисленні скалярного добутку з молодших розрядів можна використовуючи *метод формування макродобутоків на базі попередніх обчислень* [13-16]. Під час реалізації даного методу попередні обчислення виконуються на базі однорозрядних суматорів в процесі введення множених  $A_j$ . Під час формування часткового добутку для двох і більше пар добутоків використовуються попередні обчислення, кількість яких залежить від пар добутоків, для яких формується частковий добуток. Кількість додаткових однорозрядних суматорів визначається таким виразом:

$$h = \frac{m}{v} 2^v - (v + 1), \quad (0.6)$$

де  $v$  – кількість пар добутоків, для яких формується частковий добуток.

Формування часткового добутку для двох пар добутоків виконується за таким виразом:



$$P_{li} = \begin{cases} 0, & \text{якщо } B_{(2l-1)i} = B_{2li} = 0 \\ A_{2l-1}, & \text{якщо } B_{(2l-1)i} = 1, B_{2li} = 0 \\ A_{2l}, & \text{якщо } B_{(2l-1)i} = 0, B_{2li} = 1 \\ A_{2l-1} + A_{2l}, & \text{якщо } B_{(2l-1)i} = B_{2li} = 1 \end{cases}, \quad (0.7)$$

де  $l=1, \dots, n/2$ .

Використання алгоритму формування часткових добутоків за формулою (0.7) дозволяє зменшити в два рази кількість часткових добутоків і, відповідно, кількість входів багатовходового суматора. Структура пристрою, який працює за даним алгоритмом, наведена на рис. 0.2, де СМ – суматор; БлПОР1, ..., БлПОР $m/2$  – блоки попередніх обчислень і регістрів, КМ – комутатор.

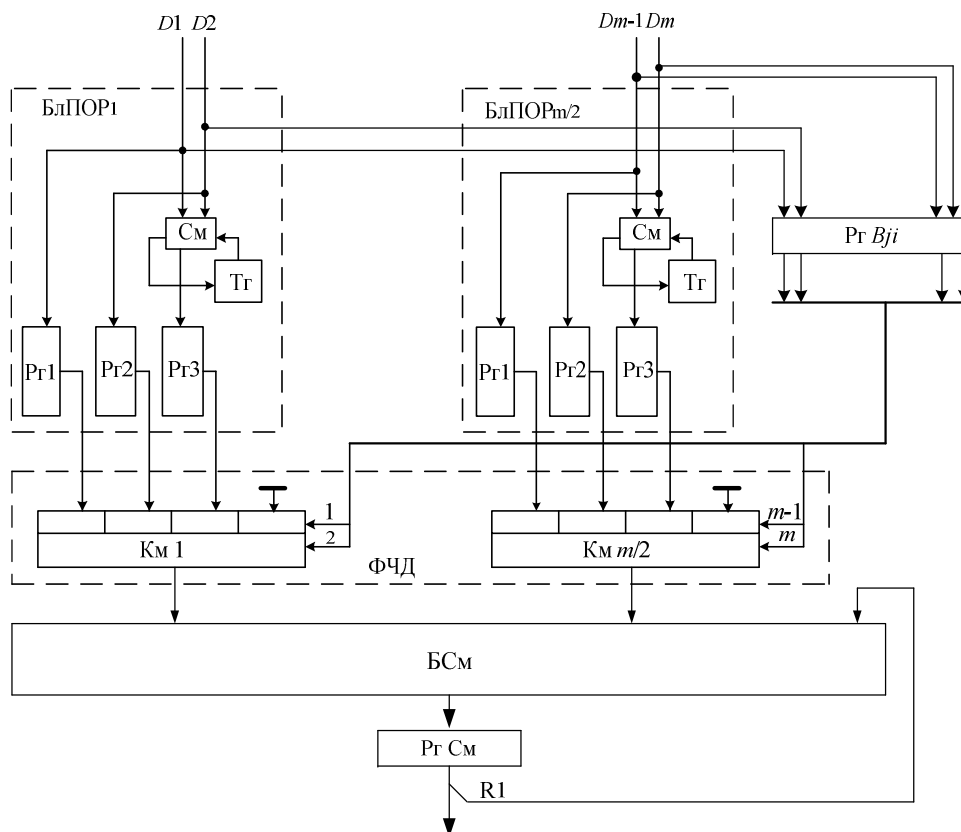


Рисунок 0.2 Структура пристрою обчислення скалярного добутку на базі попередніх обчислень.

Затрати обладнання на реалізацію пристрою обчислення скалярного добутку на базі попередніх обчислень визначаються за таким виразом:

$$W_2 = (1,5m + 1)W_{P_2} + 1,5mW_{T_2} + 0,5mW_{K_{m4-1}} + W_{(0,5m+1)-C_m} + 0,5mW_{C_{m1}},$$

де  $W_{P_2}$ ,  $W_{T_2}$ ,  $W_{K_{m4-1}}$ ,  $W_{(0,5m+1)-C_m}$ ,  $W_{C_{m1}}$  – затрати обладнання відповідно на регістр, тригер, чотири вхідний  $n$  розрядний комутатор,  $(0,5m+1)$ -вхідний суматор і однорозрядний суматор.

Обчислення скалярного добутку в даному пристрої виконується за  $2n$  такти, кожний з яких дорівнює:

$$t_2 = t_{P_2} + t_{K_{m4-1}} + t_{(0,5m+1)-C},$$

де  $t_{P_2}$ ,  $t_{K_{m4-1}}$ ,  $t_{(0,5m+1)-C_m}$  – час спрацювання відповідно регістра, формувача часткових добутків і  $(m+1)$ -вхідного суматора.

Збільшення кількості сум парних добутків, для яких формується частковий добуток, веде до збільшення кількості попередніх обчислень і до зменшення кількості входів багатовходового суматора. Вибір кількості сум парних добутків, для яких необхідно формувати частковий добуток, залежить, у першу чергу, від кількості сум парних добутків  $m$ .

*Таблично-алгоритмічний метод паралельно-вертикального обчислення сум парних добутків* використовується для випадку, коли множени  $A_j$  є константами. Реалізація такого методу обчислення передбачає формування часткових добутків за формулою (0.5) для кожної пари множників. Шляхом додавання часткових добутків  $P_{ji} = A_j B_{ji}$  з однаковими вагами отримуємо макрочастковий добуток  $P_i$ . Знаючи значення всіх операндів  $A_j$ , можна обчислити таблицю макрочасткових добутків  $P_i$  для всіх можливих комбінацій  $i$ -их розрядних зрізів множників  $A_j$  [9]. Кількість можливих варіантів макрочасткових добутків  $P_i$  (обсяг таблиці)

залежить від кількості пар операндів  $m$ . Обсяг таблиці макрочасткових добутоків  $P_l$  визначається так:

$$Q = 2^m. \quad (0.8)$$

Для обчислення скалярного добутку за таблично-алгоритмічним методом необхідно попередньо обчислити всі можливі комбінації макрочасткових добутоків  $P_i$  і записати їх у пам'ять таким чином, щоб комбінація значень розрядних зрізів множників  $B_j$  була адресою комірки пам'яті, в якій зберігається відповідне значення макрочасткового добутку  $P_i$ . Вибір макрочасткового добутку  $P_i$  з пам'яті здійснюється за адресою, яка є розрядним зрізом множників  $B_j$ , згідно з таким виразом [8]:

$$P_i = \begin{cases} 0, & \text{якщо } B_{1l} = B_{2l} = B_{3l} = \dots = B_{ml} = 0 \\ A_1, & \text{якщо } B_{1l} = 1, B_{2l} = B_{3l} = \dots = B_{ml} = 0 \\ A_2, & \text{якщо } B_{1l} = 0, B_{2l} = 1, B_{3l} = \dots = B_{ml} = 0 \\ A_1 + A_2, & \text{якщо } B_{1l} = 1, B_{2l} = 1, B_{3l} = \dots = B_{ml} = 0 \\ \vdots & \\ A_2 + A_3 + \dots + A_m, & \text{якщо } B_{1l} = 0, B_{2l} = B_{3l} = \dots = B_{ml} = 1 \\ A_1 + A_2 + \dots + A_m, & \text{якщо } B_{1l} = B_{2l} = B_{3l} = \dots = B_{ml} = 1 \end{cases}. \quad (0.9)$$

Основними етапами реалізації таблично-алгоритмічного паралельно-вертикального методу обчислення оператора скалярного добутку є [12,13]:

- обчислення таблиці всіх можливих макрочасткових добутоків  $P_i$ ;
- запис в пам'ять обчислених значень макрочасткових добутоків  $P_i$ ;
- зчитування макрочасткових добутоків  $P_i$  з комірок пам'яті за адресою, яка відповідає значенню розрядних зрізів множників  $B_j$ ;
- підсумовування макрочасткових добутоків  $P_i$  із зсувом накопиченої суми вправо на один розряд.

Процес обчислення скалярного добутку  $Z$  доцільно здійснювати з молодших розрядів, що забезпечить зменшення розрядності накопичувального суматора.

Структура пристрою, який використовує таблиці макрочасткових добутків  $P_i$  для обчислення оператора скалярного добутку, подана на рис. 0.3, де Рг – реєстр; ПерК – перетворювач кодів; ФМЧД – формувач макрочасткових добутків; ОЗП – оперативний запам'ятовуючий пристрій; БК – блок керування; СВ – суматор-віднімач; ША – шина адреси; ШД – шина даних; ШУ – шина управління; ШФ – шинний формувач; У1, У2, У3, У4, У5, У6, У7 – сигнали управління відповідно режимом роботи ПерК, режимом роботи ШФ1, записом/читанням в ОЗП, вибіркою ОЗП, режимом роботи СВ, скидом РгСм та режимом роботи ШФ2.

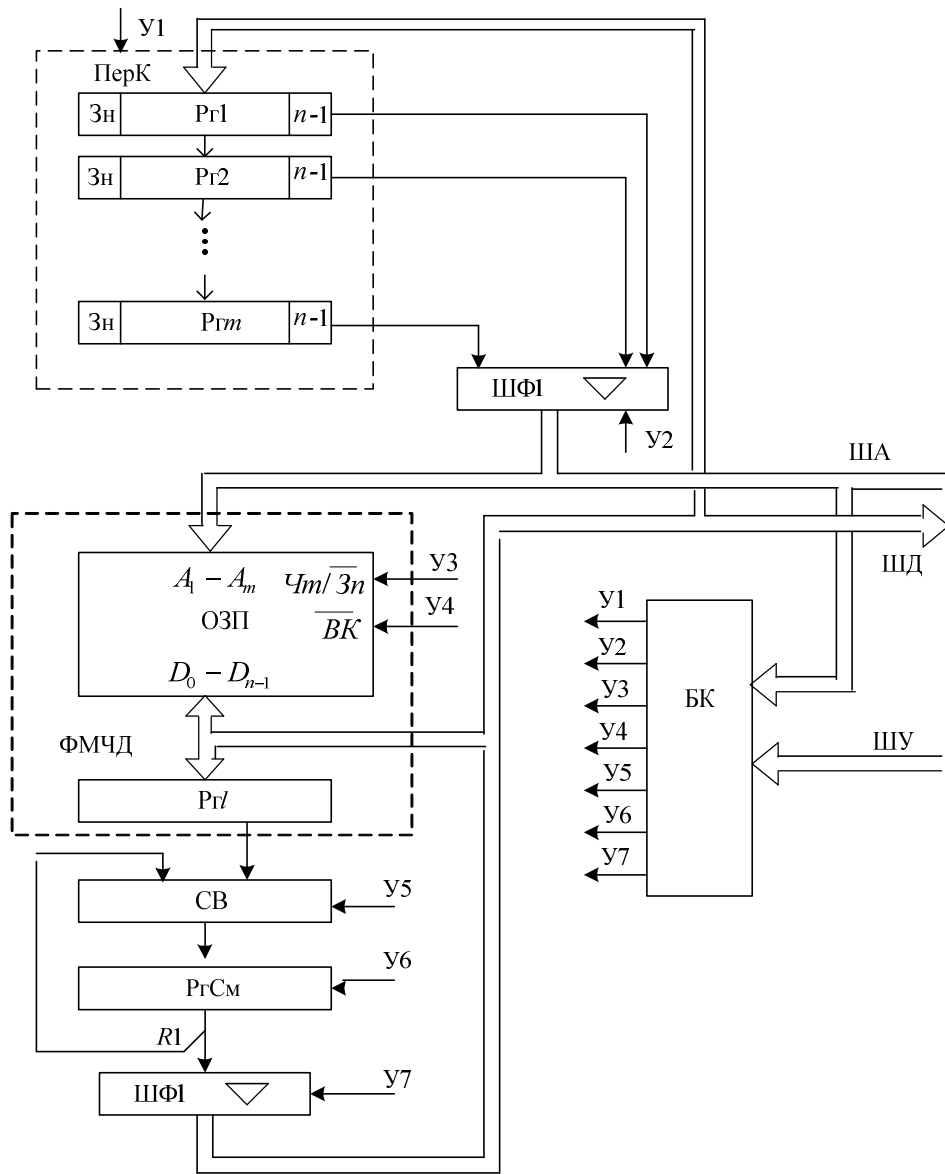


Рисунок 0.3. Структура таблично-алгоритмічного пристрою обчислення скалярного добутку.

Апаратна реалізація вдосконаленого вертикального багатооперандного методу обчислення оператора скалярного добутку  $Z$  передбачає попереднє обчислення всіх можливих варіантів макрочасткових добутків  $P_i$  з подальшим записом їх в ОЗП. Крім того, перед початком обчислення оператора скалярного добутку  $Z$  значення множників  $B_j$  в паралельному коді зберігаються в регістрах перетворювача ПерК. Сигнал лог. 1 на вході  $Y_1$  перетворювача ПерК встановлює регістри  $P_{r1}$ - $P_{rm}$  в режим зсуву в сторону молодших розрядів, а

сигнал лог. 1 на вході У2 шинного формувача ШФ1 дозволяє передачу розрядних зрізів множників  $X_j$  з виходів регістрів Рг1-Рг $m$  на адресні входи ОЗП.

Затрати обладнання на реалізацію таблично-алгоритмічного пристрою обчислення скалярного добутку визначаються за таким виразом:

$$W_3 = (m + 2)W_{P_2} + 2W_{ШФ} + W_{ОЗП} + W_{СВ} + W_{БК},$$

де  $W_{P_2}$ ,  $W_{ШФ}$ ,  $W_{ОЗП}$ ,  $W_{СВ}$ ,  $W_{БК}$  – затрати обладнання відповідно на регістр, шинний формувач, ОЗП, суматор-віднімач і блок керування.

Обчислення оператора скалярного добутку  $Z$  у запропонованому пристрої здійснюється по тактах. У кожному такті роботи пристрою здійснюється зчитування з ОЗП макрочасткового добутку  $P_i$ , який зберігається за адресою, що відповідає значенню розрядного зрізу  $B_{1i}, \dots, B_{mi}$  у відповідності з формулою (0.9). Кожний  $i$ -ий макрочастковий добуток  $P_i$  додається до зсунутої на один розряд вправо раніше накопиченої суми, що надходить з виходів регістра РгСм. Оскільки пристрій працює з числами в додатковому коді, то макрочастковий добуток  $P_1$ , який відповідає значенню знакового розрядного зрізу  $B_{13n}, \dots, B_{m3n}$ , віднімається від раніше накопиченої суми. Кількість тактів, необхідних для отримання результату обчислення скалярного добутку  $Z$  за алгоритмом з аналізом одного розряду множників  $B_{1i}, \dots, B_{mi}$ , визначається їх розрядністю.

Період конвеєрного такту, з яким працює пристрій, визначається за формулою:

$$t_3 = t_{ШФ} + t_{ОЗП} + t_{P_2},$$

де  $t_{ШФ}$ ,  $t_{P_2}$ ,  $t_{ОЗП}$  – час спрацювання відповідно шинного формувача ШФ, регістра Рг і ОЗП.

Зменшити конвеєрний такт роботи пристрою можна шляхом використання більш швидкодіючої елементної бази.

Пряма реалізація даного методу доцільна для  $m \leq 16$ , оскільки зазвичай використовується 16-розрядна шина адрес, і ємність пам'яті буде дорівнювати 216n. При  $m \geq 16$  використовуються декілька блоків пам'яті, кількість яких дорівнює  $\left\lceil \frac{m}{16} \right\rceil$ , де  $\lceil \cdot \rceil$  – знак округлення до більшого цілого. Для отримання результату додатково необхідно використовувати багатовходовий суматор, кількість входів у якому також визначається за формулою  $\left\lceil \frac{m}{16} \right\rceil$ .

*Паралельно-вертикальний метод обчислення скалярного добутку з надходженням операндів старшими розрядами вперед* забезпечує суміщення процесів приймання і обчислення. Обчислення за даним методом описується такою формулою [10]:

$$Z = \sum_{j=1}^m A_j B_j = \sum_{j=1}^m \sum_{i=0}^{n-1} 2^{-i} [(A_{j0} A_{j1} \dots A_{ji}) B_{ji} + (B_{j0} B_{j1} \dots B_{j(i-1)}) A_{ji}]. \quad (0.10)$$

Зробивши у формулі (0.10) зміну черговості додавання, отримаємо:

$$Z = \sum_{j=1}^m A_j B_j = \sum_{i=0}^n 2^{-i} \sum_{j=1}^m [(A_{j0} A_{j1} \dots A_{ji}) B_{ji} + (B_{j0} B_{j1} \dots B_{j(i-1)}) A_{ji}]. \quad (0.11)$$

Якщо у формулі (0.11) вираз  $\sum_{j=1}^m [(A_{j0} A_{j1} \dots A_{ji}) B_{ji} + (B_{j0} B_{j1} \dots B_{j(i-1)}) A_{ji}]$ , який є сумою часткових добутків, замінити на макрочастковий добуток  $P_i$ , то обчислення скалярного добутку буде мати такий вигляд:

$$Z_i = 2Z_{i-1} + P_i, \quad (0.12)$$

де  $Z_i$  – сума накопичених часткових результатів;  $Z_0 = 0$ .

У даному методі обчислення формування часткових добутків для кожної пари операндів відбувається за таким виразом:

$$P_{ji} = \begin{cases} 0, & \text{коли } B_{ji} = A_{ji} = 0 \\ A_{j0}A_{j1}\dots A_{ji} & \text{коли } B_{ji} = 1, A_{ji} = 0 \\ B_{j0}B_{j1}\dots B_{j(i-1)} & \text{коли } A_{ji} = 1, B_{ji} = 0 \\ A_{j0}A_{j1}\dots A_{ji} + B_{j0}B_{j1}\dots B_{j(i-1)} & \text{коли } B_{ji} = A_{ji} = 1 \end{cases} . \quad (0.13)$$

Обчислення результату  $Z$  здійснюється за алгоритмом, який передбачає зсув вліво суми часткових результатів  $Z_{i-1}$  при нерухомому макрочастковому добутку  $P_i$ .

Структура пристрою, який реалізує алгоритм обчислення скалярного добутку з надходженням операндів послідовним кодом старшими розрядами вперед, наведена на рис. 0.4, де ФЧР – формувач часткових результатів, ТІ – тактові імпульси.



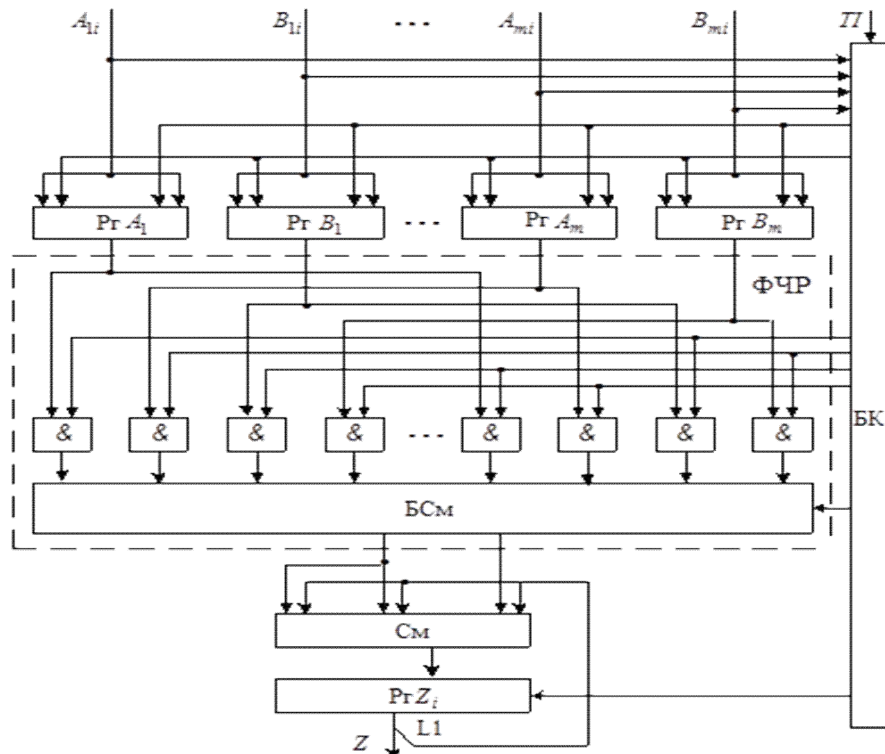


Рисунок 0.4. Структура пристрою обчислення скалярного добутку з надходженням операндів послідовним кодом старшими розрядами вперед.

У даному пристрої надходження чисел  $A_j$  і  $B_j$  здійснюється одночасно послідовним кодом старшими розрядами вперед. Розряди чисел  $A_1, \dots, A_m$  і  $B_1, \dots, B_m$  записуються в регістрах  $Pr A_1, Pr B_1, \dots, Pr A_m, Pr B_m$  і надходять у ФЧР, на виходах якого формується частковий результат обчислення  $P_{pi}$  скалярного добутку. Формування часткових результатів обчислення  $P_{pi}$  здійснюється за допомогою логічних елементів І та багатовходового суматора БСм. Отримання результату обчислення скалярного добутку  $Z_i$  у даному пристрої виконується за алгоритмом, який передбачає зсув вліво суми часткових результатів  $Z_{i-1}$  при нерухомому частковому результаті  $P_{pi}$  у відповідності з формулою  $Z_i = 2Z_{i-1} + P_{pi}$ . Обчислення скалярного добутку в даному пристрої виконується за  $n$  тактів, тривалість яких визначається так:

$$t_4 = t_{P_2} + t_{2m-Cm},$$

де  $t_{2m-C_m}$  – час додавання  $2m$   $n$ -розрядних чисел.

Апаратні затрати на реалізацію пристрою обчислення скалярного добутку з надходженням операндів послідовним кодом старшими розрядами вперед визначаються так:

$$W_4 = 2(m+1)W_{P_2} + 2mnW_i + W_{2m-C_m} + W_{C_m2n} + W_{BK},$$

де  $W_i$ ,  $W_{2m-C_m}$ ,  $W_{C_m2n}$ ,  $W_{BK}$  – апаратні затрати на реалізацію відповідно елемента I,  $2m$ -вхідного суматора,  $2n$ -розрядного суматора і блоку керування.

Послідовне формування макрочасткових добутків передбачає одночасне послідовно-порозрядне надходження всіх операндів і формування в кожному такті відповідних частин макрочасткових добутків. Кількість макрочасткових добутків залежить від кількості розрядів множників, які аналізуються для їх отримання. Визначається кількість макрочасткових добутків за формулою:

$$r = \left\lceil \frac{n+1}{g} \right\rceil, \quad (0.14)$$

де  $r$  – кількість груп, на які розбиваються множники  $B_j$ ;  $n$  – розрядність множників  $B_j$ ;  $g$  – кількість розрядів у групі, які аналізуються для отримання макрочасткових добутків.

Процес формування макрочасткових добутків доцільно здійснювати з молодших розрядів, що забезпечить зменшення розрядності накопичувального суматора. Вираз, за яким виконується формування  $P_l$  макрочасткового добутку, де  $l=1, \dots, r$ , має такий вигляд [16]:

$$P_l = \sum_{h=1}^s 2^{-(h-1)} \sum_{j=1}^m P_{jlh}, \quad (0.15)$$

де  $s$  – розрядність  $P_{jl}$  часткового добутку для  $j$ -ї пари операндів.

Сам  $P_{jl}$  частковий добуток обчислюється так:

$$P_{jl} = A_j K_{jl}, \quad (0.16)$$

де  $K_{jl}$  –  $l$ -й коефіцієнт при множеному  $A_j$  для формування часткового добутку для  $j$ -ї пари операндів.

Кількість коефіцієнтів  $K_j$  і необхідна кількість додаткових однорозрядних суматорів  $d$ , які потрібні для формування  $P_{jl}$  часткового добутку, визначається за такими формулами [2]:

$$K = 2^g + 1, \quad d = 2^{g-2} - 1. \quad (0.17)$$

За формуванням часткових добутків алгоритми обчислення скалярного добутку можна розділити на дві групи: з прямим формуванням і з формуванням на базі попередніх обчислень з використанням додаткових суматорів. Для алгоритмів з прямим формуванням часткових добутків  $g \leq 2$ , а для алгоритмів на базі попередніх обчислень  $g \geq 3$  [2-4].

Обчислення скалярного добутку на базі даного методу здійснюється за такою формулою:

$$Z = \sum_{j=1}^m A_j B_j = \sum_{l=1}^r 2^{g(l-1)} \sum_{h=1}^s 2^{-(h-1)} \sum_{j=1}^m P_{jlh}, \quad (0.18)$$

де  $P_{jlh}$  –  $h$ -й розряд  $P_{jl}$  часткового добутку.

Основними етапами паралельно-вертикального методу обчислення скалярного добутку з послідовним формуванням макрочасткових добутків є:

- формування розрядів часткових добутків для пари операндів;
- отримання частин макрочасткових добутків шляхом паралельного однорозрядного додавання розрядів часткових добутків пар операндів;
- формування макрочасткових добутків шляхом підсумовування частин макрочасткових добутків із зсувом на один розряд вправо;
- утворення результату обчислення скалярного добутку шляхом підсумовування макрочасткових добутків із зсувом вправо на кількість розрядів, які аналізуються для отримання часткових добутків пар операндів.

Структура пристрою обчислення скалярного добутку з послідовним формуванням макрочасткових добутків подана на рис. 0.5, де Рг – регістри; ФЧД – формувач часткових добутків пар операндів; БСм – однорозрядний  $m$ -вхідний суматор; ФМЧД – формувач макрочасткових добутків; См – суматор; ПЕ – процесорний елемент.

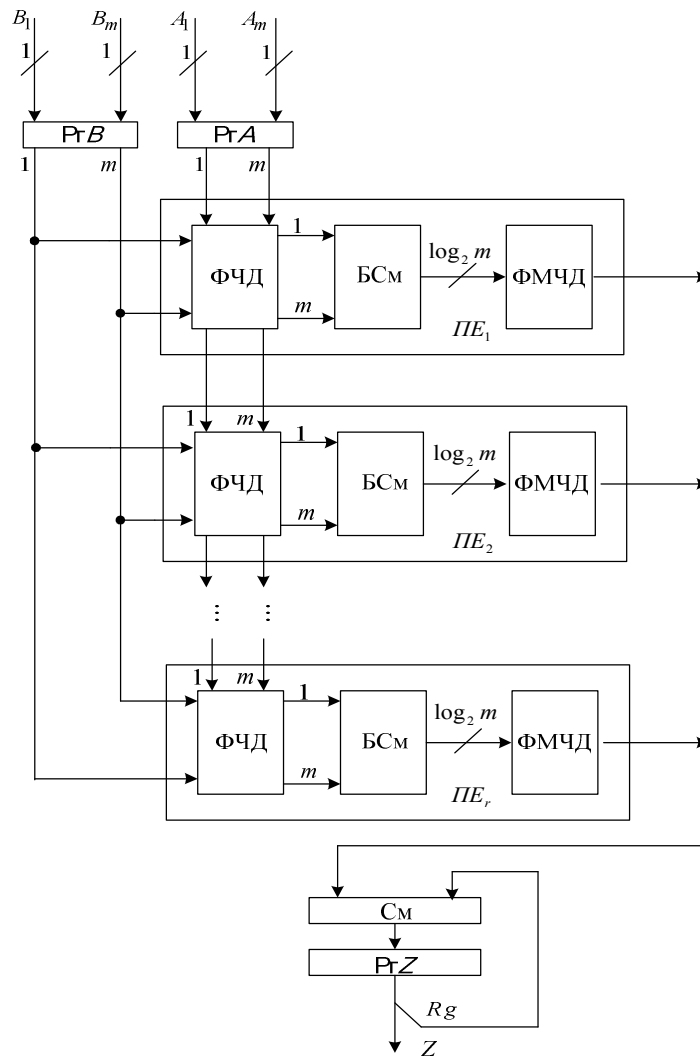


Рис. 0.5. Структура пристрою обчислення скалярного добутку з послідовним формуванням макрочасткових добутків.

У кожному такті роботи на входи пристрою надходять розрядний зріз операндів  $A_{ji}$  і  $B_{ji}$ , які записуються в регістри відповідно  $PrA$  і  $PrB$ . Дані з виходів регістрів  $PrB$  надходять на входи всіх ПЕ, а з виходів  $PrA$  – тільки на вхід  $PE_1$ . В  $PE_l$  за допомогою ФЧД формуються розряди часткових добутків пар операндів шляхом множення операндів  $A_j$  на коефіцієнти  $K_{jl}$ , які визначається при аналізі  $l$ -х груп із  $g$  розрядів операндів  $B_j$ . На  $j$ -у виході ФЧД отримуємо  $h$ -і розряди часткового добутку для  $j$ -ї пари операндів, які обчислюються так:

$$P_{jl} = A_j K_{jl},$$

де  $K_{jl}$  – коефіцієнт, який визначається при аналізі  $l$ -х груп із  $g$  розрядів операнда  $B_j$ .

Паралельним додаванням  $h$ -х розрядів часткових добутоків на БСм отримуємо  $h$ -у частину макрочасткового добутку  $P_{lh}$ , яка надходить на входи ФМЧД, де виконується підсумовування за такою формулою:

$$P_{lk} = 2^{-1} P_{l(k-1)} + P_{lh},$$

де  $k=1, \dots, s$ ,  $P_{l0} = 0$ .

Отримані на виходах ПЕ макрочасткові добутки за допомогою СМ підсумовуються у відповідності до формули:

$$Z_l = 2^{-g} Z_{l-1} + P_l,$$

де  $Z_0=0$ .

Пристрій паралельно-вертикального обчислення скалярного добутку працює за конвеєрним принципом і орієнтований на опрацювання неперервних потоків даних. Конвеєрний такт роботи такого пристрою визначається так:

$$t_5 = t_{P_2} + t_{Cmn},$$

де  $t_{P_2}$  – час спрацювання регістра,  $t_{Cmn}$  – час додавання  $n$ -розрядних чисел. Обчислення скалярного добутку здійснюється за  $n$  конвеєрних тактів.

Апаратні затрати на реалізацію пристрою обчислення скалярного добутку з послідовним формуванням макрочасткових добутоків, який працює за алгоритмом з аналізом одного розряду, визначаються так:

$$W_4 = 3W_{P_2} + nW_{PE} + W_{Cmn}$$

де  $W_{PE}$ ,  $W_{Cmn}$  – апаратні затрати на реалізацію відповідно процесорного елемента і  $n$ -розрядного суматора.

Основними компонентами розроблених структур є: тригери, регістри, суматори, суматори-віднімачі, багатовходові суматори, комутатори, ОЗП, шинні формувачі та логічні елементи. Оскільки розроблені структури пристроїв обчислення сум парних добутків орієнтовані на НВІС-реалізацію, то за одиницю вимірювання витрат обладнання візьмемо логічний вентиль, який є елементом типу інвертор, І, АБО, а для оцінки часових параметрів – величину затримки логічного вентиля  $\tau$ . Для оцінки характеристик розроблених структур пристроїв обчислення сум парних добутків використаємо дані про витрати та величину затримки основних компонентів (табл. Таблица 0.1), які наведені в [2].

Таблица 0.1. Витрати обладнання та швидкодія компонентів пристроїв обчислення скалярного добутку

№ з/п	Найменування компонента	Витрати обладнання (вентилів)	Кількість каскадів затримки ( $\tau$ вентилів)
1	тригер	$\beta 6$	$\lambda 3$
2	регістр	$\beta 7n$	$\lambda 3$
3	шинний формувач	$\beta 7$	$\lambda 3$
4	однорозрядний суматор	$\beta 18$	$\lambda 7$
5	$n$ -розрядний суматор	$\beta 20n$	$\lambda 7 \log_2 n$
6	$n$ -розрядний суматор-віднімач	$\beta 23n$	$\lambda 8 \log_2 n$
7	$m$ -вхідний $n$ -розрядний суматор-віднімач	$\beta (m-1)20n$	$\lambda 7 \log_2 n \log_2 m$
8	$m$ -вхідний $n$ -розрядний комутатор	$\beta 3mn$	$\lambda m$
9	$m$ -вхідний $n$ -розрядний	$\beta 2^m 3n$	$\lambda (m+3)$

	ОЗП		
--	-----	--	--

Для оцінки витрат і швидкодії компонентів пристроїв обчислення скалярного добутку, які реалізовані за різними технологіями, використовуються перевідні коефіцієнти для витрат  $\beta$  і швидкодії  $\lambda$ . На підставі цих даних для кожної з розроблених структур запишемо аналітичні вирази розрахунку витрат обладнання та часу обчислення. Аналітичні вирази розрахунку витрат обладнання та часу обчислення скалярного добутку для розроблених структур у залежності від кількості  $m$  і розрядності  $n$  операндів подані в табл.2.

Таблиця 0.2 Аналітичні вирази для розрахунку основних характеристик пристроїв обчислення скалярного добутку

№ з/п	Витрати обладнання (вентилів)	Час обчислення ( $\tau$ вентилів)
1	$W_1 = 28mn + 7n + 6m$	$t_1 = 2n(7 \log_2 n \log_2 m + 4)$
2	$W_2 = 20,5mn + 13n + 18m$	$t_2 = 2n(7 \log_2 n \log_2 0,5m + 7)$
3	$W_3 = (3n2^{16} + 20n) \left\lceil \frac{m}{16} \right\rceil + 7mn + 51n$	$t_3 = n(m + 9)$
4	$W_4 = 56mn + 54n$	$t_4 = n(7 \log_2 n \log_2 2m + 3)$
5	$W_5 = 41(mn + n) + 7n^2$	$t_5 = 2n(7 \log_2 n + 3)$

Графіки залежності витрат обладнання та часу обчислення для розроблених структур обчислення скалярного добутку у залежності від кількості пар добутків та їхньої розрядності подано, відповідно, на рис. 0.6 та рис. 0.7



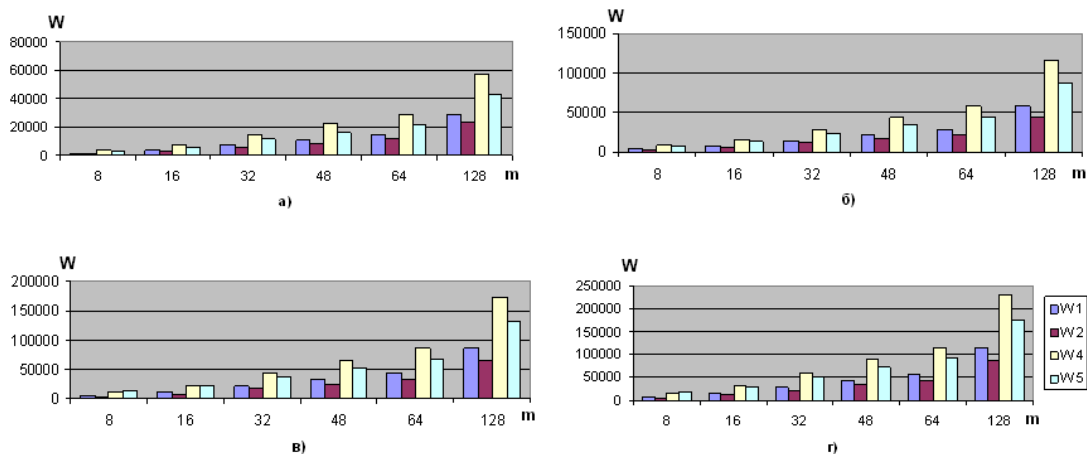


Рисунок 0.6. Графіки залежності витрат обладнання пристроїв обчислення скалярного добутку від кількості пар добутків для а)  $n=8$ ; б)  $n=16$ ; в)  $n=24$ ; г)  $n=32$ .

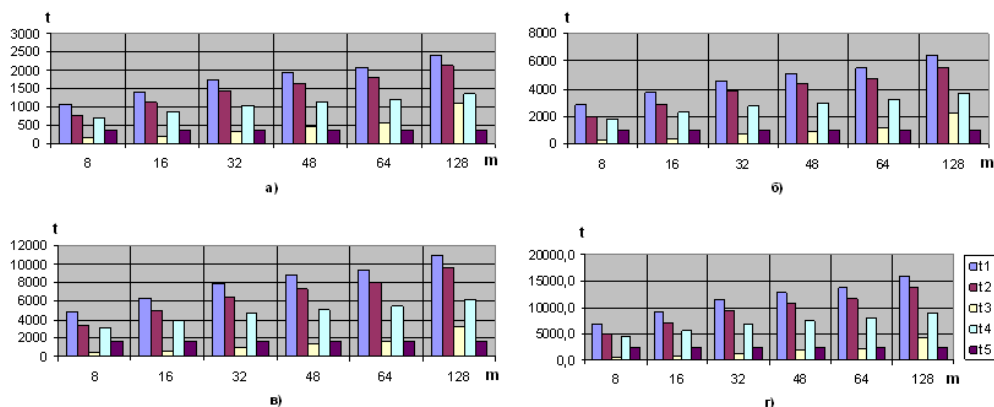


Рисунок 0.7. Графіки залежності часу обчислення скалярного добутку від кількості пар добутків для а)  $n=8$ ; б)  $n=16$ ; в)  $n=24$ ; г)  $n=32$ .

Витрати обладнання для структури 3 (рис. 0.3) подаються на окремому графіку, оскільки її робота базується на пам'яті.

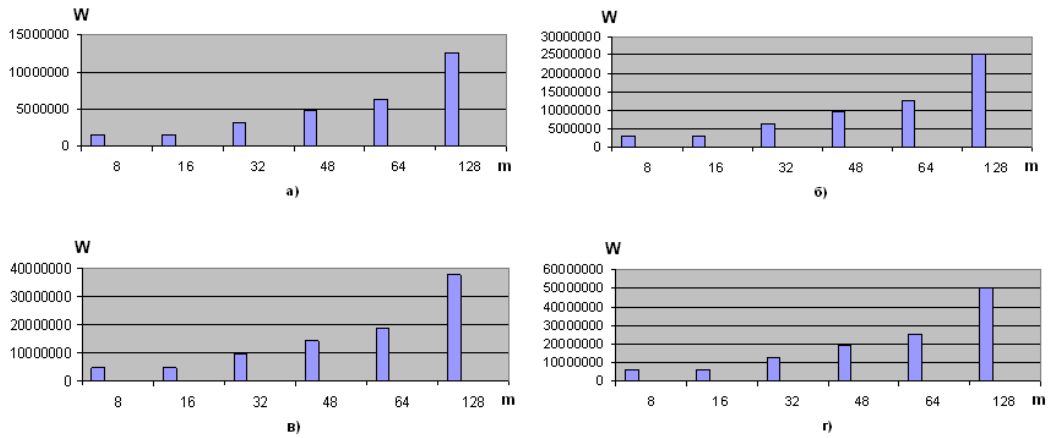


Рисунок 0.8. Графіки залежності витрат обладнання для пристрою 3 обчислення скалярного добутку від їх кількості для а)  $n=8$ ; б)  $n=16$ ; в)  $n=24$ ; г)  $n=32$ .

Як видно з графіків, найменші затрати обладнання має структура 2 (рис. 0.2), а найбільші – структура 4 (рис. 0.4). Найбільшу швидкість має структура 3 (рис. 0.3), а найменшу – структура 1 (рис. 0.1).

### 3.2 Методи та структури апаратних компонентів для реалізації групового підсумовування

У загальному випадку макрооперація групового підсумовування має наступний вигляд:

$$Z = \sum_{j=1}^M C_j, \quad (0.19)$$

де  $M$  – кількість доданків;  $C_j$  –  $j$ -й доданок [1].

Нехай доданки  $C_j$  є двійковими  $n$ -розрядними додатними числами меншими за одиницю, які записуються так:

$$C_j = \sum_{i=1}^n 2^{-i} C_{ji} . \quad (0.20)$$

Підставивши значення (0.20) у формулу (0.19), отримаємо:

$$Z = \sum_{j=1}^M \sum_{i=1}^n 2^{-i} C_{ji} . \quad (0.21)$$

Формула (0.21) відображає горизонтальну модель обчислення оператора групового підсумовування. У роботі [2] розглянуті всі можливі варіанти реалізації горизонтальної моделі групового підсумовування. Найшвидшим варіантом реалізації горизонтальної моделі групового підсумовування є паралельно-паралельний метод обчислення, граф алгоритму якого поданий на рисунку 0.9.

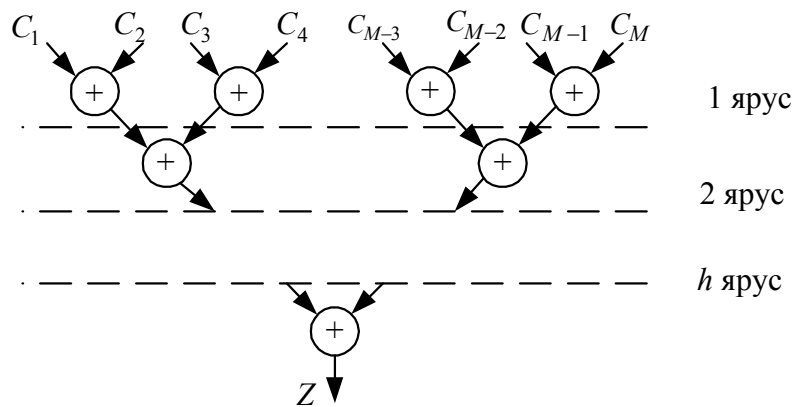


Рисунок 0.9 Граф алгоритму паралельно-паралельного групового підсумовування.

Алгоритм паралельно-паралельного групового підсумовування є каскадним. Час обчислення суми макрооперації групового підсумовування за

таким алгоритмом залежить від висоти графа (кількості ярусів), яка обчислюється так:

$$h = \lceil \log_2 M \rceil,$$

де  $\lceil \cdot \rceil$  – операція округлення до більшого цілого числа.

У кожному ярусі операнди розбиваються на пари, для кожної з яких обчислюється сума. Загальна кількість операцій додавання для обчислення суми макрооперації групового підсумовування рівна:

$$U = \frac{M}{2} + \frac{M}{4} + \frac{M}{8} + \dots + 1 = M - 1.$$

Підвищити швидкодію обчислення, ефективність використання та орієнтувати структуру макрооперації групового підсумовування на НВІС-реалізацію можна шляхом використання вертикального та багатооперандного додавання [3-7].

У нейромережах для ефективного обчислення в реальному часі оператора групового підсумовування доцільно використовувати вертикальний та багатооперандний підходи для його реалізації. При використанні даних підходів процес обчислення оператора групового підсумовування розглядається як виконання єдиної операції, що ґрунтується на базовій операції додавання значень бітів розрядного зрізу, тобто зводиться до вертикальної моделі обчислення. Замінивши у формулі (0.21) порядок підсумовування переходимо до вертикальної моделі обчислення оператора групового підсумовування, яка записується так:

$$Z = \sum_{i=1}^n 2^{-i} \sum_{j=1}^{M_i} C_{ji}, \quad (0.22)$$

де  $M_i$  – кількість доданків у  $i$ -у розрядному зрізі.

Існуючі вертикальні методи обчислення операції групового підсумовування зводять процес обчислення до перетворення багаторядного коду в однорядний. Таке перетворення ґрунтується на базовій операції перетворення трирядного коду в дворядний:

$$E = \begin{cases} C_{(j-1)1} \dots C_{(j-1)(n-1)} C_{(j-1)n} \\ + \\ C_{j1} \dots C_{j(n-1)} C_{jn} \\ + \\ C_{(j+1)1} \dots C_{(j+1)(n-1)} C_{(j+1)n} \end{cases} = \begin{cases} 0S_1 \dots S_{n-1} S_n \\ + \\ P_0 P_1 \dots P_{n-1} 0 \end{cases} \quad (0.23)$$

Перетворення трирядного коду в дворядний здійснюється за допомогою шару однорозрядних суматорів, що не мають зв'язків між собою. Для зменшення часу перетворення багаторядного коду в однорядний шари однорозрядних суматорів необхідно об'єднати за принципом дерева Уоллеса [3, 4]. Кількість шарів однорозрядних суматорів для обчислення оператора групового підсумовування визначається за формулою:

$$K = \lceil \log_{1.5} 0,5M \rceil. \quad (0.24)$$

Обчислення оператора групового підсумовування за таким методом розглядається як виконання єдиної операції, де одиниці переносів враховуються тільки один раз при заключному етапі перетворення дворядного коду в однорядний.

Пришвидшення процесу перетворення багаторядного коду в однорядний пропонується здійснити шляхом використання для перетворення наступних однорозрядних операцій:

$$\begin{aligned}
E_{3-2} &= \left\{ \begin{array}{l} C_{ji} \\ + \\ C_{(j+1)i} \\ + \\ C_{(j+2)i} \end{array} \right\} = \left\{ \begin{array}{l} P_{i-1} \\ + \\ S_i \end{array} \right\}, \\
E_{7-3} &= \left\{ \begin{array}{l} C_{ji} \\ + \\ C_{(j+1)i} \\ + \\ C_{(j+2)i} \\ + \\ C_{(j+3)i} \\ + \\ C_{(j+4)i} \\ + \\ C_{(j+5)i} \\ + \\ C_{(j+6)i} \end{array} \right\} = \left\{ \begin{array}{l} P_{i-2} \\ + \\ S_{i-1} \\ + \\ S_i \end{array} \right\}, \\
E_{15-4} &= \left\{ \begin{array}{l} C_{ji} \\ + \\ C_{(j+1)i} \\ + \\ C_{(j+2)i} \\ + \\ C_{(j+3)i} \\ + \\ C_{(j+4)i} \\ + \\ C_{(j+5)i} \\ + \\ C_{(j+6)i} \\ + \\ C_{(j+7)i} \\ + \\ C_{(j+8)i} \\ + \\ C_{(j+9)i} \\ + \\ C_{(j+10)i} \\ + \\ C_{(j+11)i} \\ + \\ C_{(j+12)i} \\ + \\ C_{(j+13)i} \\ + \\ C_{(j+14)i} \end{array} \right\} = \left\{ \begin{array}{l} P_{i-3} \\ + \\ S_{i-2} \\ + \\ S_{i-1} \\ + \\ S_i \end{array} \right\} \quad (0.25)
\end{aligned}$$

де  $E_{3-2}$ ,  $E_{7-3}$  і  $E_{15-4}$  – результати однорозрядних операцій додавання відповідно трьох, семи і п'ятнадцяти операндів. Для реалізації таких операцій використовуються 3-, 7- і 15-входові однорозрядні суматори.

Аналітичні вирази для реалізації 7-входового однорозрядного суматора запишуться так:

$$\begin{aligned}
S_2(2^0) &= Y_0L_1 \vee Y_1L_0 \vee Y_0L_3 \vee Y_1L_2 \vee Y_2L_1 \vee Y_3L_0 \vee Y_1L_4 \vee Y_2L_3 \vee Y_3L_2 \vee Y_3L_4, \\
S_1(2^1) &= Y_0L_2 \vee Y_1L_1 \vee Y_2L_0 \vee Y_2L_1 \vee Y_3L_0 \vee Y_2L_4 \vee Y_3L_3 \vee Y_3L_4 \vee Y_0L_3 \vee Y_1L_2, \\
P(2^2) &= Y_0L_4 \vee Y_1L_3 \vee Y_2L_2 \vee Y_3L_1 \vee Y_1L_4 \vee Y_2L_3 \vee Y_3L_2 \vee Y_2L_4 \vee Y_3L_3 \vee Y_3L_4,
\end{aligned} \quad (0.26)$$

$$\text{де } Y_0 = \overline{C_7 C_6 C_5}; \quad Y_1 = \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 C_5}; \quad Y_2 = \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 C_5};$$

$$Y_3 = C_7 C_6 C_5; \quad L_0 = \overline{C_4 C_3 C_2 C_1}; \quad L_1 = \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1};$$

$$L_2 = \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1};$$

$$L_3 = \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 C_1}; \quad L_4 = C_4 C_3 C_2 C_1.$$

На основі аналітичних виразів (0.26) синтезуємо комбінаційний 7-входовий однорозрядний суматор, схема якого подана на рис. 0.10.

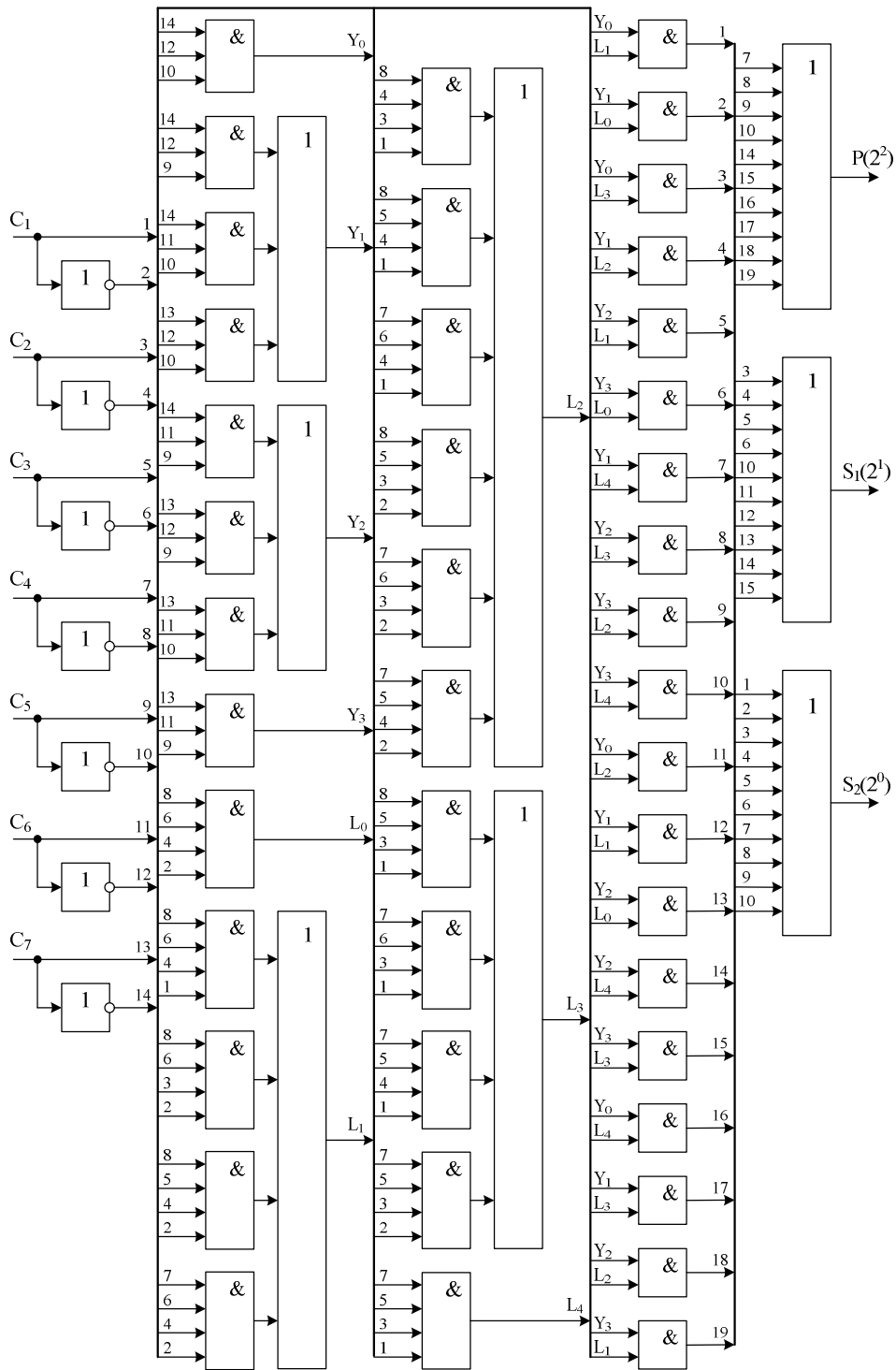


Рисунок 0.10. Схема комбінаційного 7-входового однорозрядного суматора.

Швидкодія комбінаційного 7-входового однорозрядного суматора визначається часом затримки проходження даних з входу на вихід і обчислюється так:

$$t_{C_{M7-3}} = 5t_{\log.1},$$



де  $t_{\text{лог.}I}$  – час спрацювання логічного елемента  $I$ .

Розробку аналітичного виразу для синтезу 15-входового однорозрядного суматора будемо здійснювати поетапно.

На *першому етапі* розробки розбиваємо вхідні дані на чотири групи таким чином:  $L=C_1 C_2 C_3 C_4$ ,  $Y=C_5 C_6 C_7$ ,  $K=C_8 C_9 C_{10} C_{11}$ ,  $D=C_{12} C_{13} C_{14} C_{15}$ .

На *другому етапі* для кожної групи формуємо аналітичні вирази для визначення кількості одиниць у групі:

— група  $L$

$$\begin{aligned} L_0 &= \overline{C_4 C_3 C_2 C_1}; & L_1 &= \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 C_2 \overline{C_1}} \vee \overline{C_4 C_3 \overline{C_2} C_1} \vee \overline{C_4 \overline{C_3} C_2 C_1}; \\ L_2 &= \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 C_3 \overline{C_2} C_1} \vee \overline{C_4 \overline{C_3} C_2 C_1} \vee \overline{C_4 C_3 C_2 \overline{C_1}} \vee \overline{C_4 \overline{C_3} C_2 \overline{C_1}} \vee \overline{C_4 C_3 \overline{C_2} \overline{C_1}}; \\ L_3 &= \overline{C_4 C_3 C_2 C_1} \vee \overline{C_4 \overline{C_3} C_2 C_1} \vee \overline{C_4 C_3 \overline{C_2} C_1} \vee \overline{C_4 C_3 C_2 \overline{C_1}}; & L_4 &= C_4 C_3 C_2 C_1. \end{aligned}$$

— група  $Y$

$$\begin{aligned} Y_0 &= \overline{C_7 C_6 C_5}; & Y_1 &= \overline{C_7 C_6 C_5} \vee \overline{C_7 C_6 \overline{C_5}} \vee \overline{C_7 \overline{C_6} C_5}; & Y_2 &= \overline{C_7 C_6 C_5} \vee \overline{C_7 \overline{C_6} C_5} \vee \overline{C_7 C_6 \overline{C_5}}; \\ Y_3 &= C_7 C_6 C_5. \end{aligned}$$

— група  $K$

$$\begin{aligned} K_0 &= \overline{C_{11} C_{10} C_9 C_8}; & K_1 &= \overline{C_{11} C_{10} C_9 C_8} \vee \overline{C_{11} C_{10} C_9 \overline{C_8}} \vee \overline{C_{11} C_{10} \overline{C_9} C_8} \vee \overline{C_{11} \overline{C_{10}} C_9 C_8}; \\ K_2 &= \overline{C_{11} C_{10} C_9 C_8} \vee \overline{C_{11} C_{10} \overline{C_9} C_8} \vee \overline{C_{11} \overline{C_{10}} C_9 C_8} \vee \overline{C_{11} C_{10} C_9 \overline{C_8}} \vee \overline{C_{11} \overline{C_{10}} C_9 \overline{C_8}} \vee \overline{C_{11} C_{10} \overline{C_9} \overline{C_8}}; \\ K_3 &= \overline{C_{11} C_{10} C_9 C_8} \vee \overline{C_{11} \overline{C_{10}} C_9 C_8} \vee \overline{C_{11} C_{10} \overline{C_9} C_8} \vee \overline{C_{11} C_{10} C_9 \overline{C_8}}; & K_4 &= C_{11} C_{10} C_9 C_8. \end{aligned}$$

— група  $D$

$$\begin{aligned} D_0 &= \overline{C_{15} C_{14} C_{13} C_{12}}; & D_1 &= \overline{C_{15} C_{14} C_{13} C_{12}} \vee \overline{C_{15} C_{14} C_{13} \overline{C_{12}}} \vee \overline{C_{15} C_{14} \overline{C_{13}} C_{12}} \vee \overline{C_{15} \overline{C_{14}} C_{13} C_{12}}; \\ D_2 &= \overline{C_{15} C_{14} C_{13} C_{12}} \vee \overline{C_{15} C_{14} \overline{C_{13}} C_{12}} \vee \overline{C_{15} \overline{C_{14}} C_{13} C_{12}} \vee \overline{C_{15} C_{14} C_{13} \overline{C_{12}}} \vee \overline{C_{15} \overline{C_{14}} C_{13} \overline{C_{12}}} \vee \overline{C_{15} C_{14} \overline{C_{13}} \overline{C_{12}}}; \\ D_3 &= \overline{C_{15} C_{14} C_{13} C_{12}} \vee \overline{C_{15} \overline{C_{14}} C_{13} C_{12}} \vee \overline{C_{15} C_{14} \overline{C_{13}} C_{12}} \vee \overline{C_{15} C_{14} C_{13} \overline{C_{12}}}; & D_4 &= C_{15} C_{14} C_{13} C_{12}. \end{aligned}$$

На *третьому етапі* записуємо аналітичні вирази для груп  $R$  і  $H$ , які об'єднують відповідно групи  $L$  і  $Y$  та групи  $K$  і  $D$ :

$$\begin{aligned}
R_0 &= Y_0L_0; & R_1 &= Y_1L_0 \vee Y_0L_1; & R_2 &= Y_2L_0 \vee Y_1L_1 \vee Y_0L_2; & R_3 &= Y_3L_0 \vee Y_2L_1 \vee Y_1L_2 \vee Y_0L_3; \\
R_4 &= Y_3L_1 \vee Y_2L_2 \vee Y_1L_3 \vee Y_0L_4; & R_5 &= Y_3L_2 \vee Y_2L_3 \vee Y_1L_4; & R_6 &= Y_3L_3 \vee Y_2L_4; & R_7 &= Y_3L_4; \\
H_0 &= K_0D_0; & H_1 &= K_1D_0 \vee K_0D_1; & H_2 &= K_2D_0 \vee K_1D_1 \vee K_0D_2; \\
H_3 &= K_3D_0 \vee K_2D_1 \vee K_1D_2 \vee K_0D_3; & H_4 &= K_4D_0 \vee K_3D_1 \vee K_2D_2 \vee K_1D_3 \vee K_0D_4; \\
H_5 &= K_4D_1 \vee K_3D_2 \vee K_2D_3 \vee K_1D_4; & H_6 &= K_4D_2 \vee K_3D_3 \vee K_2D_4; \\
H_7 &= K_4D_3 \vee K_3D_4; & H_8 &= K_4D_4.
\end{aligned}$$

На *четвертому етапі* на основі попередніх аналітичних виразів запишемо аналітичні вирази визначення кількості одиниць у вхідних даних, що надходять зі всіх 15 входів:

$$\begin{aligned}
F_1 &= R_0H_1 \vee R_1H_0; & F_2 &= R_2H_0 \vee R_1H_1 \vee R_0H_2; & F_3 &= R_3H_0 \vee R_2H_1 \vee R_1H_2 \vee R_0H_3; \\
F_4 &= R_4H_0 \vee R_3H_1 \vee R_2H_2 \vee R_1H_3 \vee R_0H_4; & F_5 &= R_5H_0 \vee R_4H_1 \vee R_3H_2 \vee R_2H_3 \vee R_1H_4 \vee R_0H_5; \\
F_6 &= R_6H_0 \vee R_5H_1 \vee R_4H_2 \vee R_3H_3 \vee R_2H_4 \vee R_1H_5 \vee R_0H_6; \\
F_7 &= R_7H_0 \vee R_6H_1 \vee R_5H_2 \vee R_4H_3 \vee R_3H_4 \vee R_2H_5 \vee R_1H_6 \vee R_0H_7; \\
F_8 &= R_7H_1 \vee R_6H_2 \vee R_5H_3 \vee R_4H_4 \vee R_3H_5 \vee R_2H_6 \vee R_1H_7 \vee R_0H_8; \\
F_9 &= R_7H_2 \vee R_6H_3 \vee R_5H_4 \vee R_4H_5 \vee R_3H_6 \vee R_2H_7 \vee R_1H_8; \\
F_{10} &= R_7H_3 \vee R_6H_4 \vee R_5H_5 \vee R_4H_6 \vee R_3H_7 \vee R_2H_8; & F_{11} &= R_7H_4 \vee R_6H_5 \vee R_5H_6 \vee R_4H_7 \vee R_3H_8; \\
F_{12} &= R_7H_5 \vee R_6H_6 \vee R_5H_7 \vee R_4H_8; & F_{13} &= R_7H_6 \vee R_6H_7 \vee R_5H_8; & F_{14} &= R_7H_7 \vee R_6H_8; & F_{15} &= R_7H_8.
\end{aligned}$$

На *п'ятому етапі* на основі попередніх аналітичних виразів запишемо аналітичні вирази для синтезу 15-входового однорозрядного суматора:

$$\begin{aligned}
S_3(2^0) &= F_1 \vee F_3 \vee F_5 \vee F_7 \vee F_9 \vee F_{11} \vee F_{13} \vee F_{15}, \\
S_2(2^1) &= F_2 \vee F_3 \vee F_6 \vee F_7 \vee F_{10} \vee F_{11} \vee F_{14} \vee F_{15}, \\
S_1(2^2) &= F_4 \vee F_5 \vee F_6 \vee F_7 \vee F_{12} \vee F_{13} \vee F_{14} \vee F_{15}, \\
P(2^3) &= F_8 \vee F_9 \vee F_{10} \vee F_{11} \vee F_{12} \vee F_{13} \vee F_{14} \vee F_{15}.
\end{aligned}$$

Аналогічно можна розробляти аналітичні вирази для синтезу багатовходових однорозрядних суматорів з більшою кількістю входів.

Для групового підсумовування багаторозрядних чисел використовуються багатовходові однорозрядні суматори, які синтезуються за вище розробленими аналітичними виразами. Об'єднання таких суматорів за принципом дерева

Уоллеса забезпечує перетворення багаторядного коду в дворядний, який перетворюється в однорядний за допомогою паралельного суматора. На рис. 0.11 подано приклад схеми пристрою для підсумовування восьми чисел розрядністю вісім.

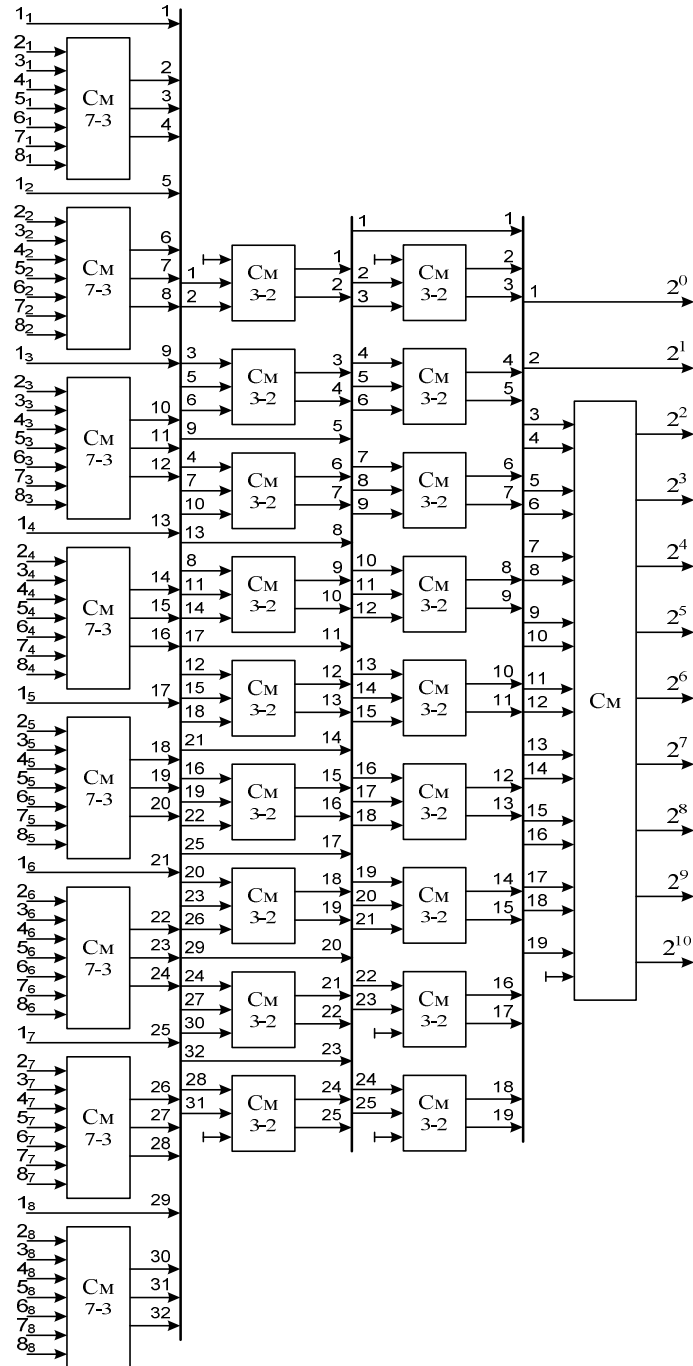


Рисунок 0.11. Схема пристрою для підсумовування восьми чисел розрядністю вісім.

Підсумовування чисел у даному пристрої здійснюється за чотири етапи шляхом використання таких перетворень: 7-рядного коду в 3-рядний, 3-рядного коду в 2-рядний та 2-рядного коду в 1-рядний код.

На першому етапі обчислення за допомогою шару з восьми 7-входових однорозрядних суматорів, які не зв'язані між собою, здійснюється перетворення 7-рядного коду в 3-рядний. На другому та третьому етапах за допомогою двох шарів, кожний з яких складається з дев'яти 3-входових однорозрядних суматорів, які не зв'язані між собою, здійснюється перетворення з 3-рядного коду в 2-рядний. На завершальному четвертому етапі за допомогою 9-розрядного паралельного суматора виконується перетворення 2-рядного коду в 1-рядний.

Час підсумовування в даному пристрої визначається за такою формулою:

$$t_{БСм} = t_{См7-3} + 2t_{См3-2} + t_{См2-1},$$

де  $t_{См7-3}$  – час перетворення 7-рядного коду в 3-рядний,  $t_{См3-2}$  – час перетворення 3-рядного коду в 2-рядний,  $t_{См2-1}$  – час перетворення 2-рядного коду в 1-рядний.

### 3.3 Метод та структура апаратного компоненту для реалізації пристрою визначення максимального числа з групи чисел

Як було описано у попередньому розділі, незалежно від типу нормалізації, яка виконується над даними перед поступленням на входи нейронної мережі, завжди потрібно знаходити максимальне значення з групи чисел.

Класично максимальне число з групи чисел визначається таким чином:

— припускається, що максимальним є перший елемент у групі чисел;

— максимальний елемент порівнюється по чергово з кожним наступним значенням;

— якщо наступне значення більше за максимальне, то воно стає максимальним і аналогічно послідовно порівнюється з наступними елементами.

Даний метод є громіздким, потребує великого об'єму обчислень і не підходить для використання в апаратних засобах реального часу. Швидкодія визначення максимального числа у ньому залежить як від розрядності чисел, так і від їхньої кількості.

Для збільшення ефективності визначення максимального числа з групи чисел доцільно застосувати паралельно-вертикальний підхід, при якому час визначення максимального числа з групи чисел не залежав від кількості чисел. При цьому у кожному такті роботи на інформаційні входи пристрою надходять розрядні зрізи всіх чисел починаючи зі старших розрядів.

Структуру такого пристрою наведено на рис. Рисунок 2., де ТІ – тактовий вхід, У – вхід початкової установки тригерів,  $x_1, \dots, x_m$  – однорозрядні інформаційні входи, де  $m$  – кількість чисел, що порівнюються, БП<sub>1</sub>, ..., БП <sub>$m$</sub>  – блоки порівняння, Т<sub>1</sub> і Т<sub>2</sub> – D-тригери, Вих – вихід результату.

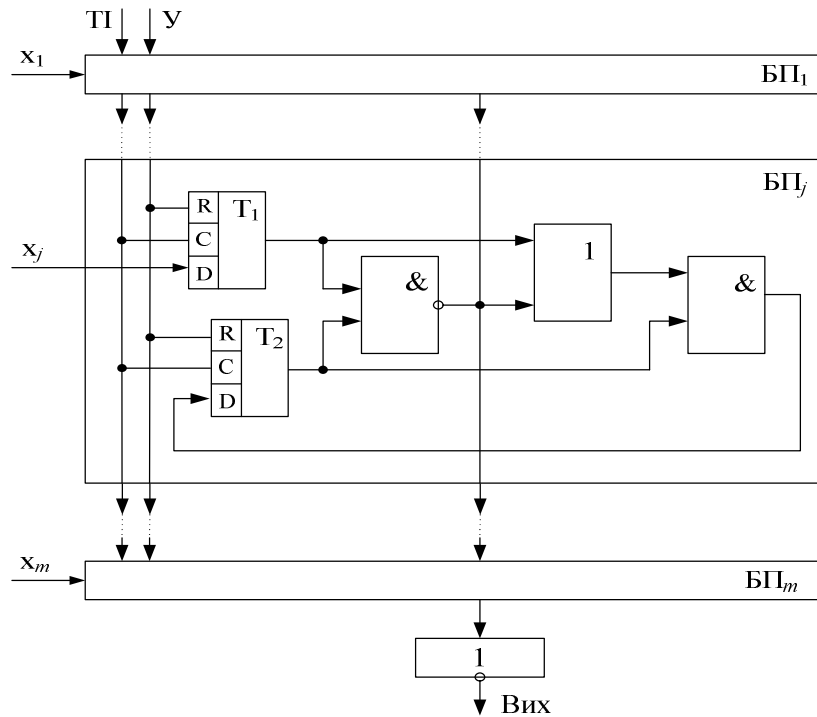


Рисунок 0.12. Структура пристрою визначення максимального числа з групи чисел.

Пристрій для визначення максимального числа з групи чисел працює наступним чином.

Перед початком роботи імпульсом початкової установки, який надходить із входу початкової установки  $У$ , тригери  $T_1$  та  $T_2$  у кожному блоці порівняння  $БП_j$  ( $j=1, \dots, m$ ) встановлюються у лог.1. Інформація з виходів тригерів  $T_1$  і  $T_2$  (лог.1) у кожному блоці порівняння  $БП_j$  встановлює на виходах елементів  $I$  сигнал лог.1.

У першому такті у кожному блоці порівняння  $БП_j$  на інформаційний вхід тригера  $T_2$  з елемента  $I$  надходить значення лог.1, а на інформаційний вхід тригера  $T_1$  з однорозрядного інформаційного входу  $x_j$  надходить значення старшого розряду  $j$ -о числа. Першим тактовим імпульсом у кожному блоці порівняння  $БП_j$  у тригер  $T_1$  записується старший розряд  $j$ -о числа, а у тригер  $T_2$  записується лог.1, яка дозволяє участь  $j$ -о числа у визначенні максимального значення. При цьому в кожному блоці порівняння  $БП_j$  значення старшого

розряду  $j$ -о числа з виходу тригера  $T_1$  поступає на перший вхід елемента  $I-HE$  з відкритим колектором та перший вхід елемента  $АБО$ . Лог.1 з виходу тригера  $T_2$  (сигнал управління) надходить на другий вхід елемента  $I-HE$  з відкритим колектором та другий вхід елемента  $I$ . Інформація з виходів елементів  $I-HE$  з відкритим колектором блоків порівняння  $БП_1, \dots, БП_m$  об'єднується по монтажному  $I$  та поступає на другий вхід елементів  $АБО$  цих блоків порівняння і на вхід елемента  $HE$ . У випадку, коли старші розряди чисел, що порівнюються, рівні нулю, то на вході елемента  $HE$  формується лог.1, а у інших випадках – лог.0. Інформація з виходу елемента  $HE$  (старший розряд максимального числа) надходить на вихід результату  $Вих$ . При лог.1 на других входах елементів  $АБО$  блоків порівняння  $БП_1, \dots, БП_m$  на їхніх виходах встановлюється сигнал лог.1, а при лог.0 – інформація з виходів тригерів  $T_1$ . Інформація з виходів елементів  $АБО$  на виходах елементів  $I$  формує сигнали управління, які надходять на інформаційні входи  $T_2$ .

Другим тактовим імпульсом інформація з однорозрядного інформаційного входу  $x_j$  (наступний розряд) записується у тригер  $T_1$  кожного блоку порівняння  $БП_j$ , а у тригер  $T_2$  записується значення з виходу елемента  $I$ , яке дозволяє (лог.1) або забороняє (лог.0) участь інформації з однорозрядного інформаційного входу  $x_j$  у подальшому формуванні максимального числа.

Формування другого і наступних розрядів результату та сигналів управління виконуються так само, як у першому такті.

Час визначення максимального числа з групи чисел у цьому пристрої залежить від розрядності чисел, і не залежить від їхньої кількості. За  $n$  тактів, де  $n$  – розрядність чисел, отримаємо максимальне число з групи  $m$  чисел.

Час обчислення максимального числа у даному пристрої рівний:

$$t_{обч} = n(t_{T_2} + t_I),$$

де  $t_{T_2}$  – час запису інформації у тригер,  $t_I$  – час затримки інформації при проходженні через логічні елементи типу АБО, І, І-НЕ.

Затрати обладнання на реалізацію даного пристрою рівні:

$$W_{Пр} = n(2W_{T_2} + 3W_I)$$

де  $W_{T_2}$  – затрати обладнання на реалізацію тригера,  $W_I$  – затрати обладнання на реалізацію логічних елементів типу АБО, І, І-НЕ.

### 3.4 Метод та апаратний компонент реалізації пристрою для обчислення суми квадратів різниць

Метод паралельно-вертикального обчислення суми квадратів різниць. Для обчислення суми різниць квадратів використаємо багатооперандний вертикальний підхід, який передбачає одночасне послідовно-порозрядне надходження операндів, формування та підсумовування в кожному такті макрочасткових результатів. Вертикальний метод обчислення суми квадратів різниць вимагає, щоб операнди були представлені в порозрядному вигляді згідно з формулою [7,9,10]:

$$X = \sum_{j=1}^n 2^{-j} x_j,$$

де  $x_j$  – значення  $j$ -го розряду операнда;  $n$  – розрядність операндів.

Паралельно-вертикальне обчислення суми квадратів різниць ґрунтується на алгоритмі вертикального піднесення до квадрату:



$$X^2 = (0.01) \wedge x_1 + 2^{-1}(0.x_1 01) \wedge x_2 + 2^{-2}(0.x_1 x_2 01) \wedge x_3 + \dots + 2^{-(n-1)}(0.x_1 x_2 \dots x_{n-1} 01) \wedge x_n = \\ = \sum_{j=1}^n 2^{-(j-1)} R_j, \quad (0.27)$$

де  $R_j$  – частковий результат піднесення до квадрату, який визначається так:

$$R_j = (0.x_1 x_2 \dots x_{j-1} 01) \wedge x_j. \quad (0.28)$$

Використовуючи багатооперандний підхід і алгоритм вертикального піднесення до квадрату операцію обчислення суми квадратів різниць можна звести до формування і підсумовування макрочасткових результатів відповідно до формули:

$$y = (X_1^e - X_1^b)^2 + (X_2^e - X_2^b)^2 + \dots + (X_N^e - X_N^b)^2 = \Delta X_1^2 + \Delta X_2^2 + \dots + \Delta X_N^2 = \sum_{k=1}^m 2^{-(k-1)} P_k, \quad (0.29)$$

$$P_k = \sum_{i=1}^N R_{ki}, \quad (0.30)$$

де  $m$  – розрядність різниць  $\Delta X_i$ ;  $R_{ki}$  –  $k$ -ий частковий результат піднесення до квадрату  $i$ -ої різниці  $\Delta X_i$ .

Підставляючи значення (0.30) у формулу (0.29) зведемо обчислення суми квадратів різниць до операції багатооперандного підсумовування:

$$y = \sum_{i=1}^N \Delta X_i^2 = \sum_{k=1}^m \sum_{i=1}^N 2^{-(k-1)} R_{ki}. \quad (0.31)$$

Основними етапами паралельно-вертикального методу обчислення суми квадратів різниць є:

- одночасне послідовно-порозрядне надходження операндів  $X_i^e, X_i^b$  і обчислення модуля  $\Delta X_i$ ;
- формування часткових результатів піднесення до квадрату  $R_{ki}$ ;
- формування макрочасткового результату обчислення  $P_k$  шляхом підсумовування часткових результатів піднесення до квадрату  $R_{ki}$ ;
- отримання результату суми квадратів різниць шляхом підсумовування із зсувом вправо на один розряд макрочасткових результатів обчислення  $P_k$ .

Структура пристрою для паралельно-вертикального обчислення суми квадратів різниць.

В залежності від способу формування та підсумовування макрочасткових результатів  $P_k$  можливі такі варіанти реалізації пристрою обчислення суми квадратів різниць:

- з послідовним формуванням і підсумовуванням макрочасткових результатів  $P_k$ ;
- з паралельним формуванням і послідовним підсумовуванням макрочасткових результатів  $P_k$ ;
- з паралельним формуванням і підсумовуванням макрочасткових результатів  $P_k$ .

При паралельно-вертикальному обчисленні суми квадратів різниць використовується паралельне формування макрочасткового результату  $P_k$  з послідовним його підсумовуванням. Структура пристрою з паралельно-вертикальним обчисленням суми квадратів різниць наведена на рис. 0.13, де Від – віднімач, Тг – тригер, Рг – регістр, ПК – перетворювач кодів, БСм – багатовходовий суматор, См – суматор, Бл – блок формування часткових результатів піднесення до квадрату.

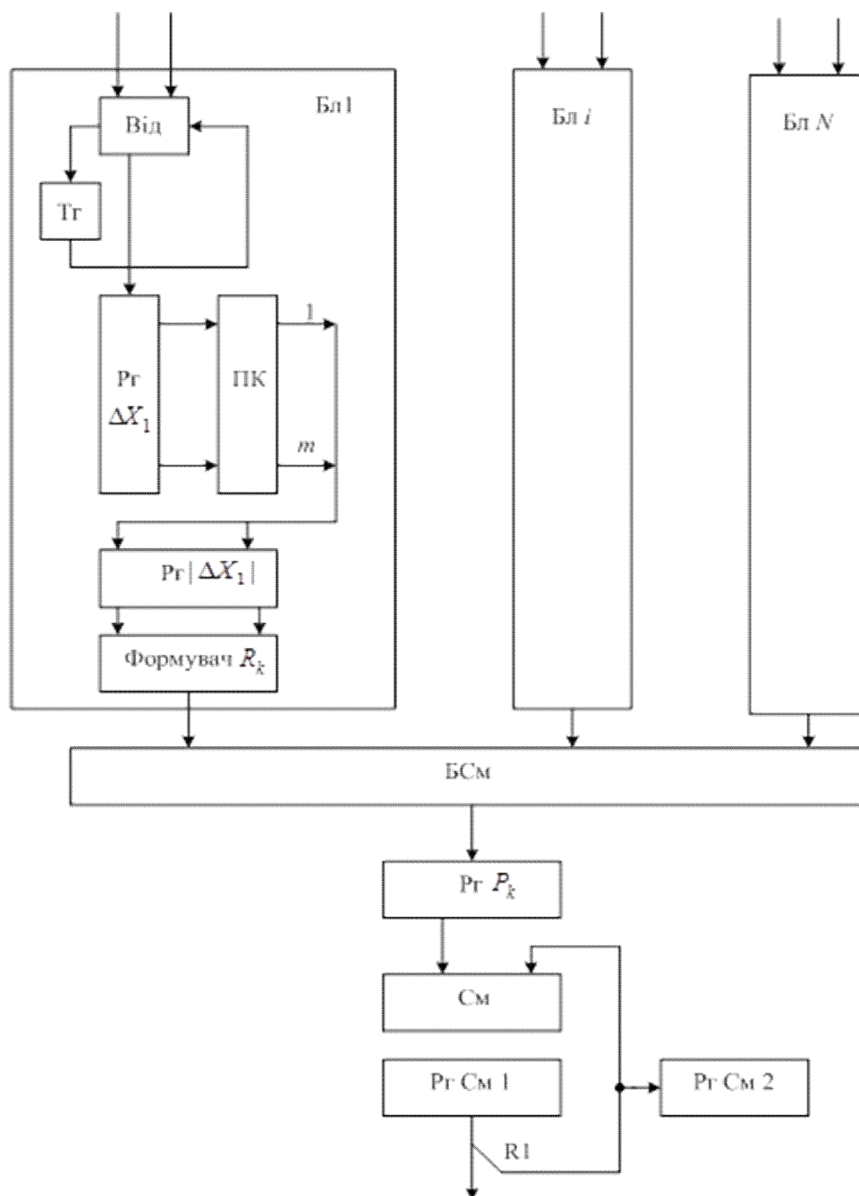


Рисунок 0.13. Структура пристрою паралельно-вертикального обчислення суми квадратів різниць.

У даному пристрої надходження чисел  $X_i^e$  і  $X_i^b$  здійснюється одночасно послідовним кодом молодшими розрядами вперед. В кожному блоці  $\text{Бл } i$  за допомогою віднімача  $\text{Від}$  обчислюється різниця  $\Delta X_i$ , яка записується в регістр  $\text{Pг } \Delta X_i$ . Обчислена різниця  $\Delta X_i$  надходить на входи перетворювача кодів  $\text{ПК}$ , на виході якого отримуємо її модуль  $|\Delta X_i|$ . У наступних тактах роботи в кожному блоці  $\text{Бл } i$  формувачем  $R_k$  формується частковий результат піднесення до

квадрату  $R_k$ . Формування часткових результатів піднесення до квадрату  $R_k$  здійснюється порозрядно починаючи з молодших розрядів модуля  $|\Delta X_i|$  у відповідності до формули (0.28). Сформовані в блоках Бл  $i$  часткові результати піднесення до квадрату  $R_{ki}$  надходять на входи багатовходового суматора БСм, де виконується їх підсумовування. Отримана сума є макрочастковим результатом обчислення суми квадратів різниць і вона записується в регістр Рг  $P_k$ . На суматорі См у відповідності з формулою (0.29) виконується додавання отриманого макрочасткового результату  $P_k$  до зсунутої вправо на один розряд накопиченої суми макрочасткових результатів.

Обчислення модуля  $|\Delta X_i|$ , формування і підсумовування макрочасткових результатів  $P_k$  елементів вибірок суміщається в часі з обчисленням  $\Delta X_i$  наступних елементів вибірок. При такому суміщенні для забезпечення режиму реального часу дані операції повинні виконуватися за час:

$$t_{об} = nT_d,$$

де  $n$  – розрядність операндів;  $T_d$  – період надходження даних.

Для обчислення суми квадратів різниць в реальному часі в даному пристрої використовується конвеєр, який повинен працювати з тактом:

$$T_k \leq \frac{t_{об}}{m+1}.$$

Необхідні апаратні затрати на реалізацію пристрою паралельно-вертикального обчислення суми квадратів різниць визначаються так:

$$W_{\Pi} = N(W_{Bid} + W_{Tz} + 2W_{Pz-m} + W_{PK-m} + W_{\Phi-T}) + W_{БСм-(N \times m)} + 2W_{Рг-(m+\log N)} + W_{См-(m+\log N)},$$

де  $W_{\text{Від}}$ ,  $W_{\text{Тз}}$ ,  $W_{P_2-m}$ ,  $W_{\text{ПК-}m}$ ,  $W_{\Phi-m}$ ,  $W_{\text{БСМ-(}N \times m)}$ ,  $W_{P_2(m+\log N)}$ ,  $W_{\text{СМ-(}m+\log N)}$  – апаратні затрати на реалізацію відповідно однорозрядного віднімача, тригера,  $m$ -розрядного регістра,  $m$ -розрядного перетворювача кодів,  $m$ -розрядного формувача часткових результатів піднесення до квадрату,  $N$ -входового  $m$ -розрядного суматора,  $(m+\log N)$ -розрядного регістра і  $(m+\log N)$ -розрядного суматора.

Основними компонентами розробленого пристрою є тригери, регістри, суматори, віднімачі, багатовходові суматори, перетворювачі кодів і формувачі часткових результатів піднесення до квадрату. Оскільки розроблений пристрій паралельно-вертикального обчислення суми квадратів різниць орієнтований на НВІС-реалізацію, то за одиницю вимірювання витрат обладнання візьмемо логічний вентиль, який є елементом типу інвертор, І, АБО. Для оцінки характеристик розробленого пристрою використаємо дані про витрати обладнання основних компонентів, які наведені в [7].

Аналітичний вираз розрахунку витрат обладнання у залежності від кількості входів  $N$  і розрядності  $m$  різниць  $\Delta X_i$  для реалізації розробленого пристрою має такий вигляд:

$$W_{\Pi} = N(24 + 55m) + 32(m + \log_2 N).$$

На основі даного аналітичного виразу будемо графік витрат обладнання (рис. 0.14) в залежності від кількості входів  $N$  і розрядності  $m$  різниць  $\Delta X_i$ .

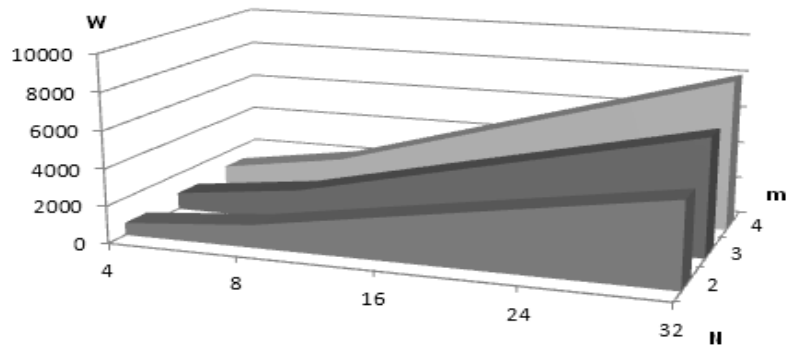


Рисунок 0.14. Графік витрат обладнання на реалізацію пристрою паралельно-вертикального обчислення суми квадратів різниць.

Як видно з графіка, витрати обладнання в більшій мірі залежать від кількості входів  $N$  і в меншій – від розрядності  $m$  різниць  $\Delta X_i$ , яка для RBF-мережі є невеликою (не перевищує 4 розряди).

## ВИСНОВКИ

Створення високоефективних спеціалізованих НВІС-структур для визначення максимальних і мінімальних чисел з масиву чисел найдоцільніше здійснювати при комплексному підході, який охоплює алгоритми, інтегральну технологію і архітектуру та враховує особливості конкретного застосування.

Особливістю алгоритмів визначення максимальних і мінімальних чисел за методом порозрядного порівняння та НВІС-структур, які їх реалізують є однотипність і простота ПЕ та локальність і регулярність зв'язків між ними.

Час визначення максимальних і мінімальних чисел з масиву чисел за методом порозрядного порівняння визначається розрядністю чисел, а не їх кількістю.

Використання спільної магістралі для формування і-х розрядів максимального і мінімального чисел зменшило час їх формування.

У роботі показано, що розробку компонентів нейромережових технологій з вертикально-паралельною обробкою даних доцільно здійснювати на основі інтегрального підходу, який охоплює: сучасні методи та алгоритми навчання та функціонування нейромереж; сучасну елементну базу; нові, орієнтовані на НВІС-реалізацію, алгоритмічні, архітектурні та схемотехнічні рішення.

Запропоновано нейромережові засоби будувати за такими принципами: змінного складу обладнання; модульності, який передбачає реалізацію компонентів у вигляді функціонально-завершених пристроїв, що мають вихід на стандартний інтерфейс; узгодженості інтенсивності надходження даних з обчислювальною здатністю апаратних засобів; конвеєризації та просторового паралелізму обробки даних; локалізації та спрощення зв'язків між елементами компонентів; спеціалізації та адаптації апаратно-програмних засобів до структури алгоритмів та інтенсивності надходження даних.

Показано, що основними шляхами підвищення ефективності використання обладнання компонентів штучних нейронних мереж є: вибір ефективних методів

та алгоритмів реалізації компонентів штучних нейронних мереж; зменшення розрядності операційних пристроїв, ємності пам'яті, кількості і розрядності каналів передачі даних; узгодження інтенсивності надходження даних з обчислювальною здатністю компонентів штучних нейронних мереж.

Визначено, що узгодження інтенсивності надходження даних з обчислювальною здатністю компонентів штучних нейронних мереж реального часу здійснюється шляхом зміни тривалості конвеєрного такту, кількості і розрядності каналів надходження даних.

Визначено основні етапи синтезу компонентів штучних нейронних мереж: вибір та розробка методів алгоритмів; визначення основних параметрів апаратних засобів; перехід від алгоритму до узгоджено-паралельної структури.

Вдосконалено метод переходу від алгоритму до структури апаратних засобів систем реального часу, основними етапами якого є: декомпозиція алгоритмів розв'язання задачі; проектування комунікацій між функціональними операторами; укрупнення функціональних операторів; планування обчислень.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Цмоць І.Г., Скорохода О.В., Кісь Я.П. Принципи розробки та методи синтезу високоефективних засобів нейромережових технологій реального часу // Науково-технічний журнал "Технічні вісті". – 2009/1(29), 2(30). – С. 87-89.
2. Цмоць І.Г., Скорохода О.В. Принципи побудови комп'ютерних нейросистем реального часу // Збірник наукових праць ЛДІНТ ім. В. Чорновола "Фізико-математичні та технічні науки". – Львів. – 2010. – №3. – С. 30-36.
3. Цмоць І.Г., Балич Б.І., Скорохода О.В. Методи та НВІС-структури пристроїв паралельно-вертикального обчислення сум парних добутків. // "Відбір і обробка інформації". – Львів. – 2011. – №33 (109). – С. 109-116.
4. Ткаченко Р.О., Цмоць І.Г., Скорохода О.В. Вертикально-паралельні методи та структури для реалізації базових компонентів нейромережових технологій реального часу // Науково-публіцистичний часопис "Технічні вісті". – Львів. – 2010. – №1-2. – С. 166-169.
5. Ткаченко Р.О., Цмоць І.Г., Скорохода О.В. Метод і НВІС-структура пристрою для обчислення суми квадратів різниць для RBF-мережі реального часу. // Вісник НУ "Львівська політехніка" "Комп'ютерні науки та інформаційні технології". – Львів. – 2011. – № 694. – С. 20-26.
6. Скорохода О.В., Цмоць І.Г., Сенах Б.І. Особливості реалізації нейромереж реального часу // Науково-публіцистичний часопис "Технічні вісті". – Львів. – 2011. – №1(33)-2(34). – С. 28-30.
7. Цмоць І.Г., Скорохода О.В., Балич Б.І. Реалізація нейронного елемента на основі попередніх обчислень. // Вісник НУ "Львівська політехніка" "Комп'ютерні науки та інформаційні технології". – Львів. – 2011. – № 710. – С. 11-18.
8. Цмоць І.Г., Скорохода О.В., Балич Б.І. Модифікований метод та НВІС-структура пристрою групового підсумовування для нейроелемента. // Вісник НУ

“Львівська політехніка” “Комп’ютерні науки та інформаційні технології”. – Львів. – 2012. – № 732. – С. 51-57.

9. Скорохода О.В., Андрієцький Б.Р., Цмоць І.Г., Ткаченко Р.О. Засоби оцінювання параметрів динамічних об’єктів на основі нейромережевого сингулярного спектрального аналізу. // Науковий вісник національного лісотехнічного університету України: Збірник науково-технічних праць. – Львів: РВВ НЛТУ України. – 2012. – Вип.22.11. – С. 362-369.

10. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – 2-е изд., стереотип. – М.: Горячая линия-Телеком, 2002. – 382 с.

11. Галушкин А.И. Нейрокомпьютеры. Кн.3.-.М; ИПРЖР,2000.-528с.

12. Осовский С. Нейронные сети для обработки информации / Пер. с польского. – М.: Финансы и статистика, 2002. – 344 с.

13. Цмоць І.Г. Інформаційні технології та спеціалізовані засоби обробки сигналів і зображень у реальному часі. – Львів: УАД, 2005.- 227с.

14. Параллельная обработка информации: в 5т./ АН УССР. Физ-мех. ин-т. – Киев: Наукова думка, 1984. – т5: Проблемно-ориентированные и специализированные средства обработки информации / А.И. Аксенов, В.В. Аристов, Е.Ю. Барзилович и др.; Под ред. Б.Н. Малиновского и Грицика В. В. – 1990. – 504с.

15. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ – Петербург, 2003. – 1104с.

16. Круглов В.В. Искусственные нейронные сети. Теория и практика / В. Круглов, В. Борисов. — М. : Горячая линия-Телеком, 2002. — 382 с.

17. Хайкин С. Нейронные сети: полный курс / Хайкин С.; пер. с англ. — [3-е изд.]. — М. : Вильямс, 2006.

18. Осовский С. Нейронные сети для обработки информации / Осовский С.; пер. с польского. — М. : Финансы и статистика, 2002. — 344 с.

19. Палагин А.В. Реконфигурируемые вычислительные системы / А. Палагин, В. Опанасенко. — К. : Просвіта, 2006. — 280 с.

20. Цмоць І. Інформаційні технології та спеціалізовані засоби обробки сигналів і зображень у реальному часі / І. Цмоць. — Львів : УАД, 2005. — 227 с.
21. Воеводин В.В. Параллельные вычисления / В. Воеводин, Вл. Воеводин. — СПб. : БХВ-Петербург, 2002. — 608 с.
22. Касьянов В.Н. Графы в программировании: обработка, визуализация и применение / В. Касьянов, В. Евстигнеев. — СПб. : БХВ-Петербург, 2003. — 1104 с.
23. Апаратна реалізація багатосарових перцептронів з неітераційним навчанням / Р. Ткаченко, П. Ткаченко, І. Цмоць // Збірник наукових праць Інституту проблем моделювання в енергетиці НАН України. — 2005. — Випуск 29. — С. 103-113.
24. Кун С. Матричные процессоры на СБИС. — М.: Мир, 1991. — 672 с.
25. Цмоць І. Г. Інформаційні технології та спеціалізовані засоби обробки сигналів і зображень у реальному часі. — Львів: УАД. — 2005. — 227 с.
26. Стрямець С. П., Цмоць І. Г. Паралельні алгоритми та НВІС-структури обчислення сум парних добутоків. // Вісник НУ “Львівська політехніка” “Комп’ютерна інженерія та інформаційні технології”. — 2003. — № 496. — С. 255–263.
27. Цмоць І. Г. Паралельні алгоритми та матричні НВІС-структури пристроїв множення для комп’ютерних систем реального часу. Науково-технічний журнал “Інформаційні технології і системи”. — 2004. — Т. 7, №1. — С. 5–16.
28. Березький О., Цмоць І. Методи та НВІС-структури для множення матриці на матрицю у реальному часі. Вісник НУ “Львівська політехніка” “Комп’ютерні системи проектування. Теорія і практика”. — 2007. — № 591. — С. 63–76.
29. Солонина А. И., Улахович Д. А., Яковлев А. А. Алгоритмы и процессоры цифровой обработки сигналов. — СПб: БХВ-СПб, 2001. — 464 с.
30. Параллельная обработка информации: Т.4. Высокопроизводительные системы параллельной обработки информации / Под ред. В. В. Грицыка. — Киев: Наук. думка, 1988. — 272 с.

31. Самофалов К. Г. и др. Прикладная теория цифровых автоматов. – К: Вища шк., 1987. – 375 с.
32. Грушицкий Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608 с.
33. Каневский Ю. С. Системные процессоры. – К: Техніка, 1991. – 173 с.
34. Шалыто А. А. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука, 2005. – 780 с.
35. Цмоць І. Г. Особливості проектування спеціалізованих комп'ютерних систем для обробки інтенсивних потоків інформації // Збірник наук. праць ІПМЕ НАН України: "Моделювання та інформаційні технології". – 1999. – Вип. 8. – С. 143–149.
36. Цмоць І. Г. Алгоритми і структури ВІС перемножувача комплексних чисел. Вісник ДУ "Львівська політехніка". – 1998. – № 237. – С. 231–240.
37. Цмоць І. Г. Алгоритмічні операційні пристрої для обчислення базових операцій алгоритмів швидкого перетворення Фур'є комплексної послідовності. // Зб. наук. пр. Інституту проблем моделювання в енергетиці НАН України. – 1999. – Вип. 2. – С. 159–173.
38. Галушкин А.И. Нейрокомпьютеры. Кн.3. – М.: ИПРЖР, 2000. – 528 с.
39. Осовский С. Нейронные сети для обработки информации / Пер. с польского. – М.: Финансы и статистика, 2009. – 344 с.
40. Березький О.М., Цмоць І.Г. Методи, алгоритми та НВІС-структури для множення матриці на вектор у реальному часі. Вісник Хмельницького національного університету, №3, 2007/ Т1. – С. 134-140.
41. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая Линия-Телеком, 2002. – 382 с.
42. Рутковская Д., Пилиньский Л., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы / Пер. с польского – М.: Горячая линия-Телеком, 2007. – 452 с.
43. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – 2-е изд., стереотип. – М.: Горячая линия-Телеком, 2002. – 382 с.

44. С. Хайкин. Нейронные сети: полный курс, 2-е изд.:Пер. с англ.. – М.:”Вильямс”, 2006.
45. Галушкин А.И. Нейрокомпьютеры. Кн.3. – М; ИПРЖР,2000. – 528с.
46. Осовский С. Нейронные сети для обработки информации / Пер. с польского. – М.: Финансы и статистика, 2002. – 344 с.
47. А.В. Палагин, В.Н. Опанасенко. Реконфигурируемые вычислительные системы. – К.: Просвіта, 2006. – 280с.
48. Кун С. Матричные процессоры на СБИС. – М.: Мир. – 1991. – 672 с.
49. Цмоць І.Г. Інформаційні технології та спеціалізовані засоби обробки сигналів і зображень у реальному часі. – Львів: УАД. – 2005. – 227 с.
50. Солонина А.И., Улахович Д.А., Яковлев А.А. Алгоритмы и процессоры цифровой обработки сигналов. – СПб: БХВ-СПб. – 2001. – 464 с.
51. Параллельная обработка информации: Т.4. Высокопроизводительные системы параллельной обработки информации /Под ред. В.В. Грицька/. – Киев: Наук. думка. – 1988. – 272 с.
52. Самофалов К.Г. и др. Прикладная теория цифровых автоматов. – К: Вища шк. – 1987. – 375 с.
53. Грушицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург. – 2002. – 608 с.
54. Каневский Ю.С. Системные процессоры. – К: Техніка. – 1991. – 173 с.
55. Шалыто А.А. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука. – 2005. – 780 с.
56. Р.О. Ткаченко, І.Г. Цмоць, О.В. Скорохода. Вертикально-паралельні методи та структури для реалізації базових компонентів нейромережових технологій реального часу. Науково-технічний журнал “Технічні вісті” 2010/1(31), 2(32). – С.166-169.
57. В.В. Грицьк, Р.О. Ткаченко, І.Г. Цмоць. Технологія нейрокомп’ютерингу реального часу. Вісник НУ “Львівська політехніка” “Комп’ютерні науки та інформаційні технології”№ 672 – Львів, 2010. – С.359-371.

58. Цмоць І.Г. Інформаційні технології та спеціалізовані засоби обробки сигналів і зображень у реальному часі. – Львів: УАД, 2005.- 227с.
59. Справочник по цифровой вычислительной технике:(Электрон. вычисл. машины системы)/Б.Н. Малиновський, В.Я. Александров. В.п. Боюн и др. Под ред. Б.Н. Малиновського. – К. : «Техніка», 1980. 320 с.
60. Цифровая обработка информации на основе быстродействующих БИС. С.А. Гамкрелидзе, А.В. Завьялов, П.П. Мальцев, В.Г. Соколов; Под ред. В.Г. Домрачева.- М.: Энероатомиздат, 1988.- 136 с.
61. Коуги П.М. Архитектура конвейерных ЭВМ.: Пер. с англ. – М.: Радио и связь, 1985. – 360 с.
62. Гамаюн В.П. О развитии многооперандных вычислительных структур / В.П. Гамаюн // Управляющие системы и машины. – 1990. - №4. – С. 31-33.
63. Ромм Я.Е. Методы обработки потока целочисленных групповых данных. Групповые арифметические операции / Я.Е. Ромм // Кибернетика и системный анализ. – 1998. – №3. – С.123-151.
64. Паулин О.Н. Модель и метод проектирования многооперандных сумматоров на базе симметрических функций / О.Н. Паулин, А.М. Ляховецкий // Тези доповідей на міжнар. конф. з індуктив. моделювання МКІМ-2002. – Львів: ДНДІ. – 2002. – С. 208-213.