Міністерство освіти і науки України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Algorithms and hardware implementation of sigmoidal activation functions of artificial neural networks

Студент гр. КІ - 21
Снігур Вадим Сергійович

Тернопіль - 2019

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

НВІС – надвелика інтегральна схема

ОП – операційними пристроями

ПЕ – процесорний елемент

ПЗ – програмне забезпечення

ФО – функціональні оператори

ПГА – потоковий граф алгоритму

БПП - багатопортова пам’ять.

CPU англ. Central Processing Unit – центральний процесор

CUDA англ. Compute Unified Device Architecture – програмно-апаратна архітектура паралельних обчислень

GPU англ. Graphics Processing Unit – графічний процесор

1 АНАЛІЗ ГАЛУЗЕЙ ЗАСТОСУВАННЯ, МЕТОДІВ ТА ЗАСОБІВ СИГМОЇДАЛЬНИХ ФУНКЦІЙ АКТИВАЦІЇ

1.1 Галузі застосування засобів сигмоїдальних функцій активації

В останні десятиліття зростає інтерес до апаратної реалізації штучних нейронних мереж (ARN). Це видно по кількості публікацій на цю тему. Це пов'язано із швидким розвитком елементної бази, що використовується при реалізації цифрового ANL (надвеликі інтегральні схеми - VLIS). Залишається одна з актуальних проблем – збільшення швидкості роботи нейронних мереж. Одним із способів пришвидшити їх роботу за допомогою одночасності є їх реалізація на ПЛІС.

Швидкість штучних нейронів залежить від функції активації сигмовидної кишки. Реалізація на FPGA вимагає багатьох апаратних ресурсів [1, 2, 3]. [1] дає огляд найважливіших методів реалізації сигмовидної функції.

У сучасних комп'ютерах високопродуктивна обробка даних досягається за допомогою просторового та часового паралелізму. Особливий інтерес представляє концепція підвищення продуктивності обчислювального пристрою шляхом наближення його структури до структури виконуваного алгоритму [1]. Перетворення алгоритму в його апаратну модель відбувається за допомогою повного апаратного відображення блок-схеми виконуваного алгоритму (PGA) операційними пристроями (OP), які виконують оператори функцій (FO) алгоритму і взаємопов'язані відповідно до плану алгоритму [2]. Пристрої, структура яких пристосована до PGA, мають наступні переваги:

Ви можете виконати операцію з усіма операндами, незважаючи на стан інших операцій (синхронізація не потрібна).

Між операціями чітко визначений обмін даними. Керування операціями здійснюється за допомогою передачі даних між ними; Алгоритм вирішується за один цикл, що прискорює конкретне завдання кілька разів.

Однак існує багато питань, в основному пов'язаних із методом створення та модифікації графіків алгоритмів для можливості їх апаратної реалізації [1, 2, 3, 4].

Враховані різні варіанти синтезу паралельних комп'ютерних сортувальних пристроїв та їх конструкція від програмного опису алгоритму до апаратної реалізації з можливістю зміни ширини паралельної форми.

Доволі високою є частка сортування серед операцій, що виконуються комп'ютером, ця задача є однією з головних проблем обробки даних, і, звичайно, розуміється як задача розміщення елементів невпорядкованого набору значень $\overline{X} = \{x_1, x_2, ..., x_N\}$ в порядку монотонного зростання або спадання $\overline{X} \approx \overline{Y} = \{(y_1, y_2, ..., y_N) : y_1 \le y_2 \le ... \le y_N\}$. Сортування на основі ПГА будуються комутуючі мережі, які забезпечують здатність до обміну даними між компонентами багатопроцесорної комп'ютерної системи [2]. Також, алгоритми сортування – є зручним засобом для визначення переваг тієї або іншої моделі та для їх порівняння між собою.

Обчислювальні зусилля процесу раціоналізації досить високі. Тому для деяких відомих простих методів (сортування бульбашок, сортування включеннями тощо) кількість необхідних операцій визначається квадратичною залежністю від кількості організаційних даних [5]. Паралельне виконання операцій алгоритму сортування кількома операційними пристроями одночасно набагато прискорює час виконання алгоритму. Серед алгоритмів сортування паралельні операції можуть виконуватися для алгоритмів, в яких порядок операцій залежить лише від кількості вхідних даних, а не від значень їх ключів (неадаптивні алгоритми) [6]. Серед алгоритмів сортування є неадаптивні: алгоритм сортування на основі методу "парної" перестановки [7], алгоритм сортування на основі модифікованого методу "бульбашки" [6], алгоритм сортування на основі методу Бутчера [5]. У процесі проектування було кілька недоліків, які стали суттєвими зі складністю проектів. Одним з мінусів є те, що вам часто доводиться виконувати ручну роботу над алгоритмами розпаралелювання або зміною ширини паралельної фігури. Звідси випливає, що така робота вимагає додаткового часу та високої кваліфікації, і все ж не гарантує підтримання простого рішення, що призводить до того, що на стадії складності проект стає збитковим. Це змушує шукати нові способи пришвидшити дизайн.

Одним з основних способів пришвидшення розробки інструментів пошуку мінімальних та максимальних елементів у наборі даних є використання нейронних мереж.

Вищезазначені методи паралельного сортування засновані на застосуванні тієї самої базової операції "порівняння та переставлення", при якій пара ключів із набору даних порівнюється для сортування та перестановки цих значень, якщо їх порядок не відповідає умовам сортування. Ці методи відрізняються лише порядком порівняння пар, а основна операція "порівнювати та переставляти" однакова для таких алгоритмів. Алгоритми даних PGA, отримані для сортування 16 вхідних значень, показані на рисунку 1.1

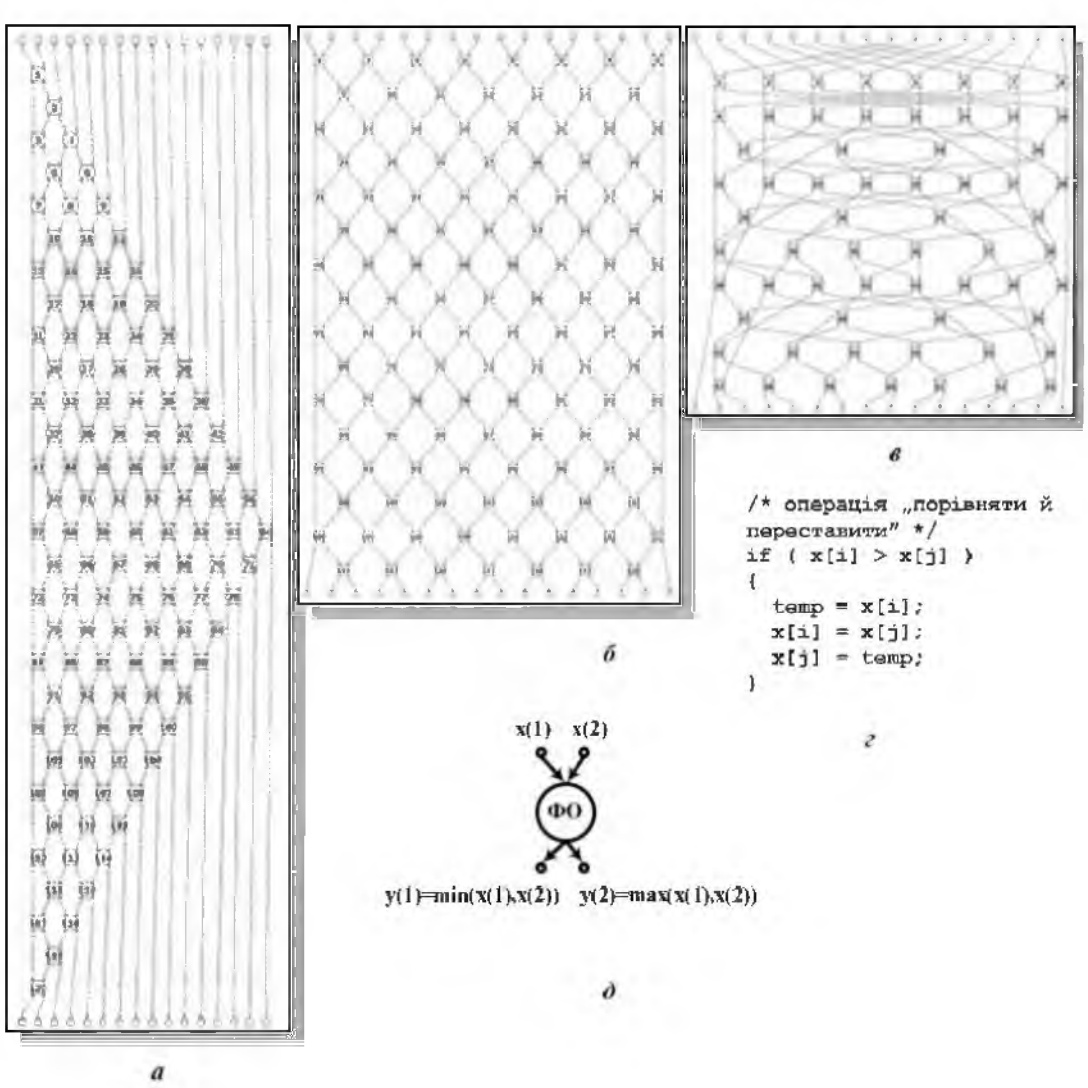


Рисунок 1.1 – Алгоритми сортування 16 значень: а – ПГА сортування модифікованим методом “бульбашки”; б – ПГА сортування “парно-непарної” перестановки; в – ПГА сортування Бетчера; г, д – базова операція “порівняти й переставити”

При однаковій кількості вхідних значень ці алгоритми мають однакову ширину PGA, але виконують різну кількість операцій "порівняння та переставлення" та мають різну кількість рівнів.

Загальна кількість FO-PGA дорівнює загальній кількості операцій "порівняння та перестановка". Для алгоритмів сортування, заснованих на методі "парно-непарної перестановки" та модифікованому методі "бульбашок" для вхідних значень, кількість FD становить [2], а для алгоритму сортування на основі методу Бетчера - що підтверджується результатами моделювання. Рівень PGA для цих алгоритмів різний і найбільший для модифікованого алгоритму сортування за методом "бульбашок" - [2]. Для алгоритму сортування „парних непарних" висота PGA дорівнює кількості вхідних значень, а для алгоритму сортування м'ясника вона найменша і однакова. Оскільки алгоритми виконують операцію "порівняння та перестановки", можна припустити, що затримка всіх FD дорівнює одиниці. Апаратною складністю цих обчислювальних пристроїв є кількість ПЗ PGA, а затримка часу - кількість рівнів PGA.

Сфери застосування нейромереж:

- Економіка та бізнес: ринковий прогноз, автоматична торгівля, оцінка ризику дефолту, прогноз неплатоспроможності, оцінка нерухомості, виявлення завищених та недооцінених компаній, автоматичний рейтинг, оптимізація портфеля, оптимізація товарних і грошових потоків, автоматичне зчитування чеків та форм, безпека транзакцій на пластикових картках.

- Медицина: обробка медичних зображень, моніторинг стану пацієнта, діагностика, факторний аналіз ефективності лікування, очищення показань приладів від шуму.

- Авіоніка: студентський автопілот, виявлення радіолокаційного сигналу, адаптивне управління сильно пошкодженими літаками.

- Зв'язок: стиснення відеоінформації, швидке кодування-декодування, оптимізація стільникових мереж та схеми маршрутизації пакетів.

- Інтернет: пошук асоціативної інформації, електронних секретарів та агентів користувачів у мережі, фільтрація інформації в push-системах, спільна фільтрація, класифікація стрічок новин, цільова реклама, цільовий маркетинг для електронної комерції.

- Автоматизація виробництва: оптимізація режимів виробничого процесу, комплексна діагностика якості продукції (ультразвук, оптика, гамма-випромінювання, ...), моніторинг та візуалізація багатовимірної інформації про доставку, запобігання надзвичайним ситуаціям, робототехніка.

- Політичні технології: аналіз та узагальнення соціологічних опитувань, прогнозування динаміки оцінки, виявлення значущих факторів, об'єктивне накопичення виборців, візуалізація соціальної динаміки населення.

Systems Системи безпеки та безпеки: системи ідентифікації, розпізнавання голосу, масове розпізнавання, розпізнавання номерних знаків, аналіз зображень в аерокосмічній галузі, моніторинг потоку інформації, виявлення підробок.

Введення та обробка інформації: обробка рукописних чеків, розпізнавання підписів, відбитків пальців та голосів. Введення фінансових та податкових документів у комп'ютер.

- Геологічна розвідка: аналіз сейсмічних даних, асоціативні методи розвідки корисних копалин, оцінка польових ресурсів.

Існує декілька широко поширених комерційних універсальних нейромережевих програмних пакетів (Statistica Neural Networks, NeuroShell, Matlab Neural Network Toolbox, NeuroSolutions, BrainMaker). Спеціалізованих, некомерційних чи розроблених вченими-дослідниками для власних потреб нейропрограмм набагато більше.

Нейронні мережі сьогодні широко поширені: нейронні мережі - це не що інше, як новий інструмент для аналізу даних. І краще за інших, щоб це могло використовувати фахівець у своїй галузі. Основною складністю ще ширшої дифузії нейротехнологій є нездатність широкого кола професіоналів сформулювати свої проблеми таким чином, щоб було можливим просте рішення нейронної мережі.

Основні цікаві особливості нейронних мереж на практиці такі:

Наявність алгоритмів швидкого навчання: нейронну мережу з сотнями вхідних сигналів і десятками тисяч опорних ситуацій можна швидко вивчити на найпростішому комп'ютері. Тому НМ мають широкий спектр застосування і вирішують складні проблеми прогнозування, класифікації або діагностики.

- Можливість роботи за наявності великої кількості неінформативних, галасливих вхідних сигналів - їх не потрібно усувати заздалегідь, нейронна мережа сама визначить їх як непридатні для вирішення проблеми і може чітко їх відхилити.

- Здатність працювати з корельованими незалежними змінними над різними типами інформації - дискретними, кількісними та якісними, що часто ускладнює статистичні методи

- Нейронна мережа може одночасно вирішувати кілька проблем з одним набором вхідних сигналів. Наявність декількох результатів дозволяє передбачити значення кількох показників.

Алгоритми навчання висувають дуже мало вимог до структури нейронної мережі та властивостей її нейронів. Отже, якщо у вас є спеціальні знання або якщо у вас є особливі вимоги, ви можете цілеспрямовано вибрати тип та властивості нейронів та нейронної мережі, скласти структуру нейронної мережі вручну з окремих елементів та визначити необхідні властивості для кожного з цих елементів.

1.2 Методи обчислення сигмоїдальних функцій активації

Оскільки всі штучні НМ засновані на концепції нейронів, зв'язків та передавальних функцій, між різними структурами або архітектурами НМ існують подібності. Більшість змін є результатом різних правил навчання. Розглянемо деякі найпопулярніші штучні нейронні мережі.

Роза Баллат Персептрон

Персептрон Розенбаллата вважається першою моделлю нейронних мереж. В основі багатьох видів ШНМ прямого поширення лежить теорія персептронів, яка є класичною.

– Одношаровий персептрон може розпізнавати найпростіші зображення. Обчислення зваженої суми сигналів вхідних елементів здійснюється окремим нейроном, який віднімає значення зсуву і передає результат через жорстку

порогову функцію, вихід якої дорівнює або дорівнює. Рішення залежить від значення вихідного сигналу:

- $+1$ - вхідний сигнал належить до класу A,
- -1 - вхідний сигнал належить до класу B.

Вирішальні області визначають, які вхідні образи будуть віднесені до класу A, які - до класу B. Перцептрон, що складається з одного нейрона, формує дві вирішальні області, які розділено гіперплощиною.

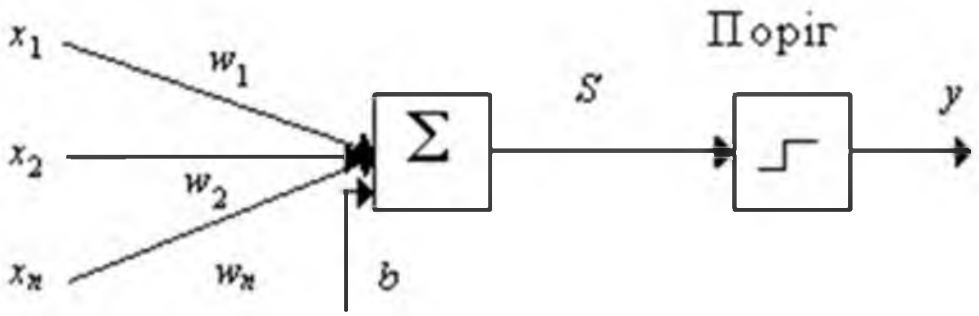


Рисунок 1.2 – Схема нейрона

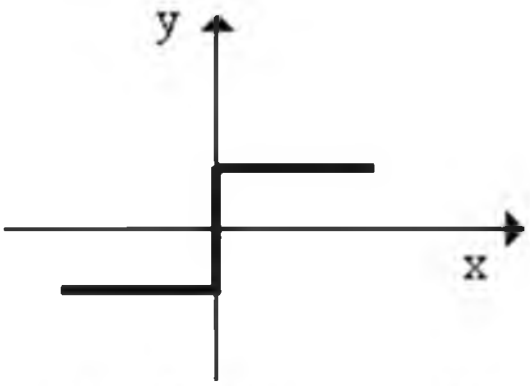


Рисунок 1.3 – Графік передатної функції

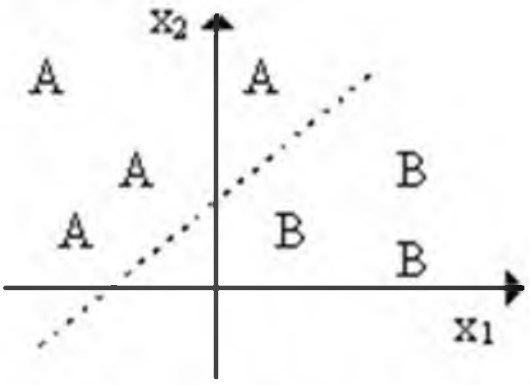


Рисунок 1.4 – Поділяюча поверхня

На рисунку показано випадок з розмірністю вихідного сигналу - 2. Інтерфейс являє собою пряму лінію на площині. Рівняння, що визначає лінію поділу, залежить від значень синаптичних ваг та переміщень.

Нейронна мережа зворотного поширення

Архітектура FeedForward BackPropagation була розроблена на початку 1970-х років кількома незалежними авторами: Werbor; Паркер; Руммелхарт, Хінтон і Вільямс. В даний час парадигма зворотного розповсюдження є популярною, ефективною та простою моделлю навчання для складних багаторівневих мереж. Їх використання в різних типах додатків породило великий клас нейронних мереж з різними структурами та методами навчання.

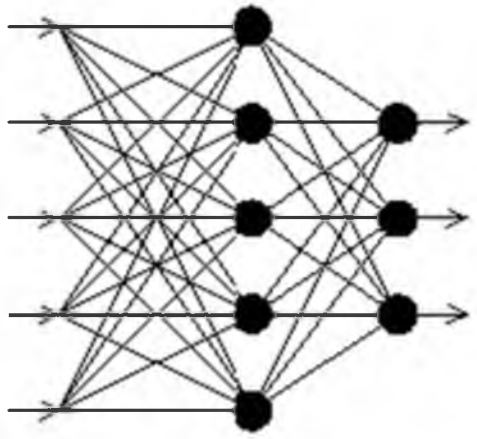


Рисунок 1.5 – Мережа зворотного поширення похибки

Типова мережа BackPropagation має вхідний шар, вихідний шар та принаймні один прихований шар. Теоретично, обмежень відносно числа прихованих шарів не існує, але практично застосовують один або два.

В шарову структуру з прямою передачею (вперед) сигналу організовано нейрони. Кожний нейрон мережі продукує зважену суму своїх входів, пропускає цю величину через передатну функцію і видає вихідне значення. Практично будь якої складності мережа може моделювати функцію, причому число шаркв і число нейронів у кожному шарі визначається складністю функції.

Визначення числа проміжних шарів і числа нейронів в них є важливим аспектом при моделюванні мережі. Більшість дослідників та інженерів використовують загальні правила, зокрема:

– Кількість входів та виходів мережі визначаються кількістю вхідних та вихідних параметрів досліджуваного об'єкту, явища, процесу, тощо.

– Якщо складність у відношенні між отриманими та бажаними даними на виході збільшується, кількість нейронів прихованого шару повинна також збільшитись.

- Якщо змодельований процес можна розбити на кілька етапів, потрібні додаткові приховані шари. Якщо процес не розбитий на етапи, додаткові шари можуть дозволити зберігання і, отже, неправильне загальне рішення.

Врахувавши всі правила та визначивши кількість шарів та кількість нейронів у кожному з них, необхідно знайти значення для синаптичних шкал та порогів мережі, які можна використовувати для мінімізації похибки отриманого результату. З цієї причини існують алгоритми навчання, в яких мережева модель адаптована до наявних навчальних даних. Пробігаючи по мережі всіх прикладів навчання та порівнюючи сформовані вихідні значення з бажаними значеннями, визначається помилка для конкретної моделі мережі. Набір помилок створює функцію помилки, яку можна розглядати як помилку мережі. Найчастіше сума квадратів помилок використовується як функція помилок.

Беручи до уваги концепцію поверхні стану, можна краще зрозуміти алгоритм навчання зворотного розповсюдження мережі. Кожне значення синаптичних ваг та порогових значень мережі (вільні параметри номера моделі) відповідає розмірності у багатовимірному просторі. Розмір відповідає несправності мережі. Для різних комбінацій ваг відповідна похибка мережі може бути представлена точкою в розмірному просторі. Ці точки утворюють поверхню - поверхню станів. Пошук найглибшої точки на багатовимірній поверхні є метою навчання нейронних мереж.

Поверхня млинів має складну структуру і досить неприємні характеристики, зокрема наявність місцевих мінімумів (точок, найнижчих у своєму конкретному середовищі, але вищих за загальний мінімум), рівних ділянок, сідлових точок та довгих вузьких ярів. Неможливо визначити

розташування глобального мінімуму на поверхні станів аналітичними засобами, тому тренування нейронної мережі є по суті вивченням цієї поверхні.

Беручи до уваги початкові конфігурації ваг та порогів (із випадково обраної точки на поверхні), алгоритм навчання поступово знаходить загальний мінімум. Розраховується вектор градієнта похибки поверхні, який вказує напрямок найкоротшого спуску на поверхню з даної точки. Щоб зменшити помилку, потрібно трохи попрацювати над нею. Зрештою, алгоритм зупиняється нижче, що може бути лише локальним мінімумом (в ідеалі глобальним мінімумом).

Складним є вибір довжини сходинок. При тривалому кроці конвергенція відбувається швидше, але існує ризик пропустити рішення або піти в неправильному напрямку. З невеликим кроком визначається правильний напрямок, але кількість ітерацій збільшується. На практиці розмір кроку приймається пропорційним крутості схилу з константою, яка є швидкістю навчання. Швидкість навчання залежить від поставленого завдання та проводиться експериментально. Ця константа також може бути функцією часу і може зменшуватися в міру просування алгоритму.

Алгоритм є ітераційним, його етапи називаються епохами. Всі приклади введення для кожної епохи подаються на вхід мережі один за одним, початкові значення мережі порівнюють з бажаними значеннями і обчислюють похибку. Значення помилок, а також градієнт поверхні станів використовуються для корекції ваг і дії повторюються. Коли минула певна кількість епох, або коли помилка досягає певного рівня незначності, або коли помилка більше не зменшується (користувач зазвичай вибирає бажаний критерій зупинки), процес навчання зупиняється.

Мережа Кохонена

На початку 1980-х років була розроблена мережа Toivo Kohonen, яка відрізняється від мереж, які були розглянуті вище. Різницяв тому, що використовується неконтрольоване навчання, а навчальний набір складається лише зі значень вхідних змінних.

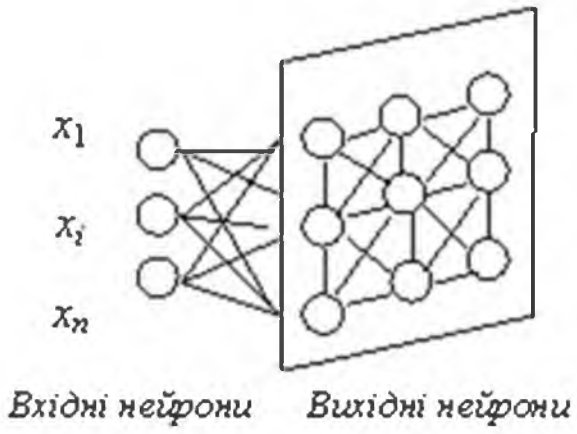


Рисунок 1.6 – Мережа Кохонена

Мережа в навчальних даних розпізнає кластери, потім дані розподіляються у відповідні кластери. Якщо наступна мережа зустріне запис, який не схожий на жоден із відомих прикладів, вона призначає його новому кластеру. Проблеми класифікації можуть бути вирішені мережею, якщо дані містять назви класів. Мережі Кохонена також можна використовувати в задачах, де класи відомі. Перевагою є здатность мережі виявляти подібність між різними класами.

Мережа Кохонена складається лише з двох шарів: вхідного та вихідного. Елементи карти знаходяться в кімнаті, як правило, двовимірній. Мережа Кохонена навчається із використанням методу послідовного наближення. Під час навчання дані надходять на входи, але мережа адаптується не до еталонного значення виводу, а до шаблонів вхідних даних. Навчання починається з випадково обраної відправної точки центрів.

При послідовному подаванні на вхід мережі навчальних прикладів визначається найбільш схожий нейрон (той, для якого скалярний добуток ваг та введення вектора мінімальний). Цей нейрон оголошено переможцем і є центром для регулювання ваги сусідніх нейронів. Це правило навчання передбачає "конкурентне" навчання з урахуванням відстані нейронів від "нейрона-переможця".

Для найбільшої схожості з вхідними даними навчання не полягає у мінімізації помилки, а в регулюванні шкал (внутрішніх параметрів нейронної мережі).

Основний ітераційний алгоритм Кохонена проходить низку епох, кожна з яких обробляє приклад із навчальної моделі. Вхідні сигнали подаються в мережу послідовно, і бажані вихідні сигнали не визначаються. Після представлення достатньої кількості вхідних векторів синаптичні ваги мережі можуть ідентифікувати кластери. Ваги організовані таким чином, що топологічно наближені вузли реагують на подібні вхідні сигнали.

Центр кластера розміщується в певному положенні на основі алгоритму, який задовольняє приклади кластера, для яких цей нейрон є "переможцем". В результаті мережевого тренінгу необхідно визначити ступінь сусідства нейронів, тобто середовища нейрона-переможця, тобто кількох нейронів, що оточують нейрон-переможця.

Спочатку велика кількість нейронів належить до навколишнього середовища, потім їх розмір поступово зменшується. Мережа утворює топологічну структуру, в якій подібні приклади утворюють групи, які розташовані близько на топологічній карті.

Мережа Хопфілда

У 1982 році Джон Хопфілд вперше презентував свою асоціативну мережу в Національній академії наук. Звідси мережева парадигма на честь Хопфілда та новий підхід моделювання називається мережею Хопфілда. Мережа Хопфілда побудована на аналогії фізики динамічних систем. Перше використання мережі включало асоціативне або орієнтоване на вміст сховище та вирішувало проблеми оптимізації.

Мережа Хопфілда використовує три рівні: вхідний рівень, рівень Хопфілда та вихідний рівень. Кількість нейронів у кожному шарі однакова. Виходи нейронів вхідного шару подаються на входи відповідних нейронів шару Хопфілда. Тут з'єднання мають фіксовану вагу. Виходи шару Хопфілда підключені до входів усіх нейронів шару Хопфілда, крім них самих, а також до відповідних елементів вихідного шару. Мережа перенаправляє дані з вхідного рівня на рівень Хопфілда, контроль відбувається під час навчання. Поки певна кількість циклів шару Хопфілда не завершиться, він коливається і поточний стан сигналів нейронів шару передається на вихідний шар. Цей статус відповідає зображенню, яке зберігається в Інтернеті.

Мережеве навчання вимагає, щоб навчальний образ подавався одночасно на вхідному та вихідному рівнях. Рекурсивний характер шару Хопфілда забезпечує спосіб виправлення всіх ваг суглобів. Відповідні пари вхід-вихід повинні бути різними для належного навчання мережі.

Коли мережа Hopfield використовується як адресація вмісту, є два основних обмеження.

Строго обмежте кількість зображень, які можна відтворити. Якщо зберігається забагато зображень, мережа може відповідати новому неіснуючому зображенню, крім усіх запрограмованих. Приблизно 15% від кількості нейронів - це обмеження ємності мережі для шару Хопфілда.

Шар Хопфілда може бути нестабільним, якщо конкретні дослідження занадто подібні. Зразок зображення вважається нестійким, якщо він застосовується в обмеженні без декомпресії, а мережа відповідає іншому зображенню з навчального набору. Вибравши більше ортогональних прикладів навчання, цю проблему можна вирішити.

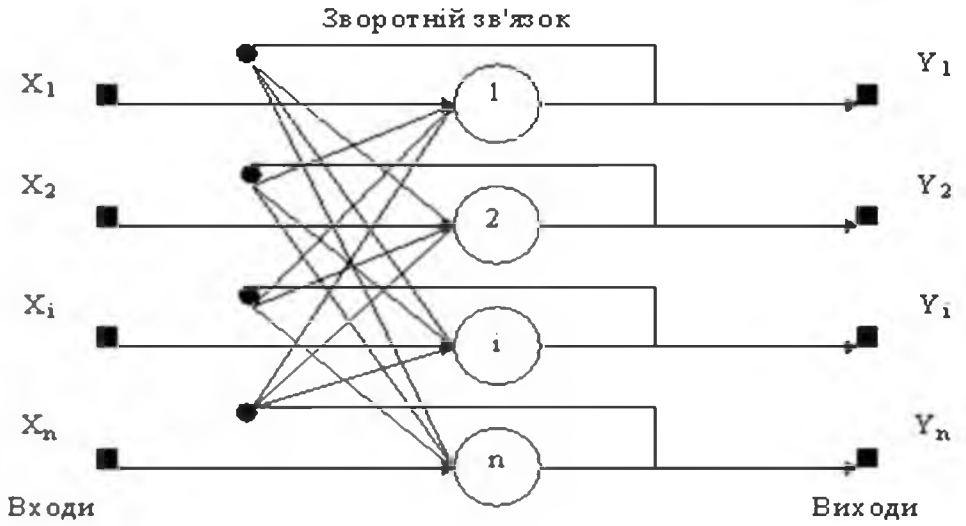


Рисунок 1.7 – Мережа Хопфілда

Є певний набір двійкових сигналів (зображень, звукових оцифровок, інших даних, які описують об'єкти або характеристики процесів) для виконання завдання асоціативної пам'яті, яка вважається зразковою. Мережа повинна бути здатною з зашумленого сигналу, поданого на її вхід, виділити ("пригадати" за

частковою інформацією) відповідний зразок або "дати висновок" про те, що вхідні дані не співпадають з жодним із зразків.

В загальному випадку, любий сигнал може бути описаний вектором $x1, xi, xn, ..., n$ - число нейронів у мережі і величина вхідних і вихідних векторів. Кожний елемент xi дорівнює або $+1$, або -1. Позначимо вектор, що описує k - ий зразок, через Xk, а його компоненти, відповідно, - $xik, k = 0,..., m-1, m$ - число зразків. Якщо мережа розпізнає (або "пригадує") якийсь зразок на основі пред'явлених їй даних, її виходи будуть містити саме його, тобто $Y = Xk$, де Y - вектор вихідних значень мережі: $y1, yi, yn$. В іншому випадку, вихідний вектор не збігається з жодний зразковим.

Якщо, наприклад, сигнали є конкретним образом зображення, то, відобразивши у графічному виді дані з виходу мережі, можна буде побачити зображення, що цілком збігається з однієї зі зразкових (у випадку успіху) або ж "вільну імпровізацію" мережі (у випадку невдачі).

Мережа Хеммінга є продовженням мережі Хопфілд. Її розробив Річард Ліппман у середині 1980-х. Мережа Хемінга реалізує найменший класифікатор помилок для бінарних вхідних векторів, де похибка визначається відстанню Хемінга. Кількість бітів, які відрізняються, визначається як відстань Хемінга. Один вхідний вектор - беззвучний приклад зображення, інший - спотворене зображення. Вихідний вектор навчального набору - це вектор класів, до яких належать зображення. У режимі навчання вхідні вектори поділяються на категорії, для яких відстань між вхідними векторами сканування та поточним вхідним вектором є мінімальною.

Мережа Хемінга створена з трьох шарів: вхідного рівня з кількістю вузлів, скільки окремих двійкових характеристик; Рівень категорій (рівень Хопфілда) із кількістю вузлів, скільки категорій або класів; вихідний рівень, який відповідає кількості вузлів на рівні категорії.

Мережа - це проста архітектура прямого розподілу з вхідним рівнем, який повністю підключений до рівня категорій. Кожен нейрон у шарі категорії зворотно зв’язаний з кожним нейроном у тому самому шарі та безпосередньо

пов'язаний із вихідним нейроном. Завдяки конкуренції формується вихід з рівня категорії до початкового шару.

Мережеве навчання Хемінга подібне до методології Хопфілда. Вхідний рівень отримує бажане навчальне зображення, а вихідний рівень вихідного рівня отримує значення бажаного класу, до якого відноситься вектор. Вихідні дані містять лише значення класу, якого стосується вхідний вектор. Рекурсивний характер шару Хопфілда забезпечує спосіб виправлення всіх ваг суглобів.

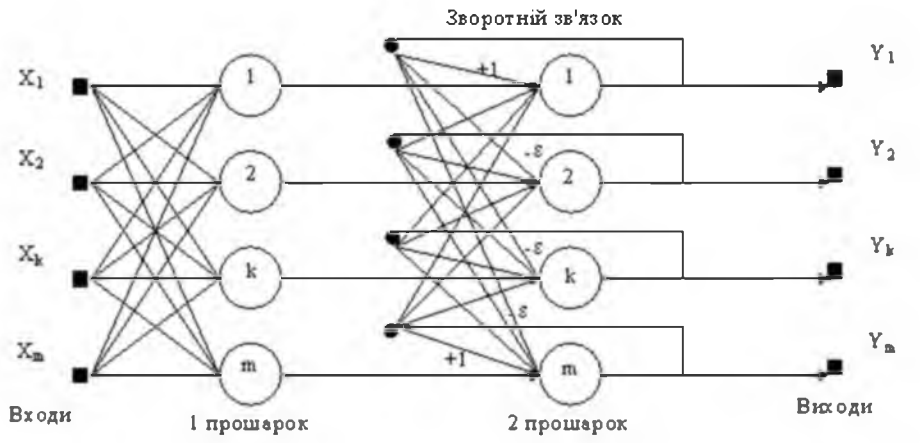


Рисунок 1.8 – Мережа Хемінга

1.3 Структури апаратних засобів сигмоїдальних функцій активації

При впровадженні обмежувальних пристроїв необхідно враховувати принципи обміну інформацією. Синхронним та асинхронним може бути обмін інформацією. Вибір принципів обміну є одним із аспектів, який окрім пропускної здатності впливає ще й на фізичну довжину каналу обміну та кількість пристроїв, які можуть бути присутніми.

Зміни даних передаються блоками від Безперервна робота під час використання за цим принципом. Передача даних базується на підключенні таймерів від передавальних та приймаючих пристроїв. Сигнали годинника визначають довгий інтервал часу для отримання інформації звіту з каналу обміну. Для запуску синхронізації використовують спеціальні сигнали. Розташування адресних сигналів та даних сигналів синхронізації визначається за

фіксованим протоколом. Для прийняття рішення, що робити далі, майже не потрібна додаткова логіка, тому синхронна передача може стати швидкою та недорогою. Обмінний канал краще підходить для синхронної передачі, підтримує високу ефективність, швидкість передачі даних та вбудований надійний механізм виявлення помилок. Основним недоліком є те, що через проблему спотворення тактових сигналів шина не може тривати довго в синхронній передачі, а інтерфейсне обладнання дорожче і складніше.

Передача запізнюється. Ви можете використовувати додаткові затримки зв'язку, щоб врахувати відсутність зв'язку між моментами доступу до відправляючого пристрою та моментами видалення готових даних. Службова інформація записується для вибору та введення початкових блоків даних. Цей принцип дозволяє поліпшити з'єднання з різними властивостями пристрою, і цей принцип забезпечує кращу стабільність. Ви можете збільшити довжину шини, незважаючи на сигнали синхронізації пероксиду. Недоліком є те, що, порівняно із синхронною передачею швидкість передачі даних через передачу великих обсягів службової інформації є нижча. Асинхронний принцип передачі даних повинен застосовуватися в системах, в яких обмін даними не є постійним і не вимагає високих швидкостей передачі даних. Аналіз способів вирішення конфліктів у пристрої, що замінює

Конфлікти можуть виникнути, якщо декілька пристроїв у системі одночасно передають дані на запасний пристрій. Є кілька способів вирішення цих конфліктів, напр. В. призначити чіткий пріоритет кожному пристрою, призначити фіксовані часові інтервали для кожного пристрою та використовувати багатопортову пам'ять.

В n-портовій пам'яті є n незалежних наборів шин адреси, даних і управління, що забезпечують одночасний і незалежний доступ до пам'яті n пристроям. Така властивість дозволяє суттєво спростити створення складних систем, але при цьому можуть виникати конфлікти і необхідно використовувати способи їх вирішення. Конфлікти виникають при звертанні декількох активних пристроїв до однієї комірки пам'яті при записі даних з кількох портів, або при записі через один порт і читанні через інші. Способи вирішення конфліктів в синхронній і асинхронній багатопортовій пам'яті (БП) відрізняються. Синхронна

БП передбачає взаємну синхронізацію активних пристроїв від одного системного таймера, тому для вирішення конфліктних ситуацій не передбачається використання додаткової логіки. В асинхронній БП для вирішення конфліктів використовується: арбітражна логіка, семафори, запити на переривання та система Master/Slave.

1.4 Постановка завдання кваліфікаційної роботи

Залежно від схем розподілу пам’яті існує кілька способів створення пристроїв, що замінюють: спільна пам’ять та спільна пам’ять. Існує дві основні моделі обміну: обмін повідомленнями (для розподіленого сховища) та використання спільного сховища.

Спільна пам’ять

Розподіляючи спільну пам'ять між усіма вхідними портами, ви можете ефективно використовувати її відповідно до типу вхідного потоку даних. У цьому випадку всі важливі вузли об'єднуються з пам'яттю через загальну шину. Перевага цього підходу полягає в тому, що обмін відбувається шляхом запису / зчитування інформації з комірок спільної пам'яті, доступних для всіх вузлів, і тому не вимагає часу для передачі даних. До недоліків можна віднести можливість конфліктів при одночасному доступі до комірки пам'яті, проблему повільного доступу до оперативної пам'яті та її обмеженої ємності, а також проблему масштабованості. Чим більше використовуваних вузлів, тим вищі витрати на розробку і менша ефективність.

Існують системи з однорідним і неоднорідним доступом до пам'яті. У системах з однорідним доступом всі вузлам надається одночасно отримувати доступ до сховища. Цей спосіб організації обміну найчастіше використовується для створення обмінних пристроїв. У системах з неоднорідним доступом до пам'яті частина спільної пам'яті виділяється кожному вузлу. Ця пам’ять має єдиний адресний простір, тому ви можете використовувати її адресу для прямого

доступу до будь-якої комірки у спільній пам’яті. Час доступу до модулів спільної пам'яті з різних вузлів різний.

Виділене сховище

У цьому випадку кожен вузол має доступ до певної фіксованої кількості виділених комірок пам'яті. Для забезпечення обміну інформацією вузли з'єднуються за допомогою каналів зв'язку. Пакет, призначений для певного вузла джерела, втрачається, якщо блок пам'яті, виділений цьому вузлу джерела, переповнюється, хоча інші блоки можуть бути порожніми на той момент. Обмін в системі здійснюється шляхом надсилання посилань та отримання повідомлень. Цю схему розподілу пам'яті використовують для завдань, які вимагають невеликого обміну даними та великої кількості пам'яті.

До переваг можна віднести масштабованість, що означає, що ви можете поєднувати велику кількість вузлів без істотного зниження ефективності їх взаємодії. Вартість системи пропорційна кількості вузлів. До недоліків можна віднести проблему обміну даними та велике споживання енергії. Обмін даними в таких системах відбувається дуже повільно в порівнянні зі швидкістю обчислень. Тому неможливо ефективно вирішити проблеми систем, які потребують інтенсивного обміну.

Постановка цілей.

Для апроксимації сигмоїдальної функції активації використано методи кусково-лінійної апроксимації (модуль FA_Sigm_N1) та апроксимацію поліномом другого порядку (модулі FA_Sigm_N2, FA_Sigm_N3). В першому методі для перемноження використовується зсуви. В другому методі використовується три перемножувачі, а в третьому тільки один перемножувач та зсуви. Виконано порівняння по точності реалізацій трьох сигмоїдальних функцій. Цифрова апаратна реалізація сигмоїдальної функції активації виконувалася мовою VHDL в середовищі розробки Quartus II. Модулі сигмоїдальних функцій реалізовані на FPGA EP3C16F484C6 сімейства Cyclone III.

2 АЛГОРИТМИ І ПРОЦЕСОРНІ ЕЛЕМЕНТИ СИГМОЇДАЛЬНИХ ФУНКЦІЙ АКТИВАЦІЇ

2.1 Формування вимог для апаратної реалізації сигмоїдальних функцій активації

Алгоритми визначення сигмоїдальних функцій активації для реалізацій VLSI повинні забезпечувати детермінований рух даних, бути добре структурованими та орієнтованими на реалізацію на наборі взаємопов'язаних процесорних елементів (РЕ). Структура та операції, що виконуються ПЕ, залежать від вимог до реалізації алгоритму. Багато взаємопов’язаних факторів необхідно враховувати одночасно при розробці або виборі алгоритмів для обчислення максимальних і мінімальних чисел.

Алгоритми повинні бути рекурсивними та локально залежними. Усі ПЕ повинні виконувати приблизно одні і ті ж операції в рекурсивному алгоритмі. Кожен з РЕ буде повторювати виконання фіксованого набору операцій над послідовністю вхідних даних. Ефективність відображення алгоритму в РЕ безпосередньо пов’язана із методом декомпозиції розв’язку задачі та перетворенням на незалежні основні операції, що реалізуються паралельно, або на залежні, що здійснюютьсяв конвеєрному режимі.

В декартовій системі координат можна сформувати точкові системи (решітки) із набору РЕ, які є моделлю паралельних апаратних структур [23]. Ця модель дозволяє дати оцінку часовій та апаратній складності алгоритму. У решітковій моделі кожен РЕ відповідає тимчасовому та просторовому j індексам, які демонструють, де і коли виконується кожна з операцій алгоритму. В алгоритмах з локальною передачею даних різниця між просторовими індексами j на кроці рекурсії обмежена деякою константою, оскільки в таких алгоритмах обмін здійснюється лише між сусідніми РЕ. Клас алгоритмів із глобальними зв’язками включає алгоритми, які мають рекурсивні просторові індекси.

Забезпечення високої швидкості є однією з головних вимог до пристроїв для вирішення максимального та мінімального числа з масиву чисел. Проблема появляється при використанні пристроїв для розв’язання проблем у режимі

реального часу, що накладає певні обмеження на процес визначення максимального та мінімального чисел з масиву чисел. Ці обмеження в першу чергу пов'язані з часом вирішення завдання, який не повинен перевищувати час обміну повідомленнями. $T_{обм}$, тобто:

$$T_p \le T_{обм} \tag{2.1}$$

Час обміну залежить як від обсягу масиву N;

розрядності n і частоти надходження вхідних даних F_d, так і від кількості k каналів та їх розрядності n_k. Такий час визначається за формулою:

$$T_{обм} = \frac{Nn}{F_d k n_k} \tag{2.2}$$

Одним з основних інтегрованих параметрів для оцінки пристроїв VLSI для обчислення максимальних та мінімальних чисел з масиву чисел є ефективність обладнання, яка бере до уваги кількість виводів інтерфейсу, однорідність структури, кількість та локальність з'єднань, продуктивність зв'язків до витрат на обладнання та дає оцінку елементам пристрою за виконанням [23]. Кількісне значення ККД обладнання визначається наступним чином:

$$E = \frac{R}{t_o(k_1 \sum_{i=1}^{s} W_{ПЕ_i} d_i + k_2 Q + k_3 Y)} \tag{2.3}$$

де R - складність алгоритму обчислення максимального та мінімального чисел;

t_o - час обчислення максимального та мінімального чисел;

$W_{ПЕ_i}$ - витрати обладнання на реалізацію i-го процесорного елемента;

d_i - кількість функціональних вузлів i-го типу;

k_1 - коефіцієнт врахування однорідності $k_1 = f(s)$;

s - кількість видів функціональних вузлів;

Q - загальна кількість зв'язків;

k_2 - коефіцієнт врахування регулярності зв'язків $k_2 = f(\Delta j)$;

Δj -просторова зв'язкова віддаіль;

Y - кількість виводів інтерфейсу;

k_3 - коефіцієнт врахування кількості виводів інтерфейсу зв'язку $k_3 = f(Y)$.

При апаратній реалізації алгоритмів визначення максимального та мінімального чисел із масиву чисел висока ефективність використання обладнання досягається узгодженням інтенсивності надходження даних $P_d = knF_d$:

де k - кількість каналів надходження даних;

n - розрядність каналів надходження даних;

F_d - частота надходження даних, із інтенсивністю обчислень (обчислювальною здатністю) апаратних засобів, яку визнають так[37]:

$$D_k = \frac{mn_m}{T_k} \tag{2.4}$$

де m - кількість каналів надходження даних у сходинках конвеєра;

n_m - розрядність каналів надходження даних у сходинках конвеєра;

T_k - такт конвеєра.

Задача розроблення пристроїв для визначення максимального та мінімального чисел із масиву, орієнтованих на НВІС-реалізацію, з високою результативністю використання обладнання зводиться до мінімізації апаратних затрат, кількості виводів інтерфейсу, збільшення однорідності структури та регулярності зв'язків при забезпечені режиму реального часу. В основу розроблення таких пристроїв доцільно покласти наступні принципи для найповнішого використання переваг сучасної НВІС-технології та забезпечення даних вимог [23-38]:

- однорідності та регулярності структури;

- локалізації та спрощення зв'язків між елементами;
- модульності побудови;
- конвеєризації та просторового паралелізму опрацювання даних;
- узгодженості інтенсивності надходження даних із інтенсивністю обчислень в пристрої;
- програмування архітектури пристрою шляхом використання програмованих логічних інтегральних схем.

2.2 Вибір принципів та елементної бази для апаратної реалізації обчислення сигмоїдальних функцій активації

Пропонується розробку нейроорієнтованих комп'ютерних систем здійснювати на основі інтегрованого підходу, який охоплює :

- сучасну елементу базу, апаратні та програмні комп'ютерні засоби;
- нейромережеві методи та алгоритми;
- обчислювальні методи, алгоритми та НВІС-структури для реалізації базових операцій нейроалгоритмів.

Підставою для деталізації моделі нейронного елемента можна вважати встановлення нових фактів в області нейрофізіології, зокрема [5]:

1. наявність декількох місць синаптичного контакту;
2. дихотомічне галуження дендритів різних порядків, що відповідає логічним операціям "І", "АБО", "Виключаюче АБО", виділення максимального або мінімального сигналу в технічних аналогах;
3. різні діаметри стовбурових дендритів, гілок, що безпосередньо прилягають до тіла нейрона, причому величина діаметра визначає степінь важливості інформації, яка проходить через дендрит;
4. наявність "доріжок" на поверхні соми, які проходять від головних стовбурних дендритів до аксона, що обумовлює наявність паралельних шляхів обробки інформації і надає можливість застосування логічних операцій над

сигналами, які надходять від різних стовбурових дендритів;

5. Особливості функціонування аксонного горбка; власне аксонний горбок встановлює передатну функцію нейрона, яка має набагато складнішу форму від прийнятих в нейромережевих технологіях сигмоїдальних або лінійних передатних функціях;

6. наявність дихотомічного галуження аксона; у вузлах галуження відбувається керування проходженням сигналу, що залежить від співвідношення діаметрів різних гілок аксона; при математичному моделюванні ці особливості можна реалізувати за допомогою логічних операцій;

7. наявність зворотного аксосоматичного зв'язку, що вже знайшло свою реалізацію при побудові рекурентних нейромереж.

Поглиблені знання відносно будови біологічного нейрона, як ефективного перетворюючого інструмента, можна розглядати як джерело базових ідей та концепцій зі створення нових парадигм нейромереж не лише на даний час, але і на віддалену перспективу.

Необхідно в основу побудови нейроорієнтованих комп'ютерних систем покласти принципи, які дозволять зменшити вартість, терміни і розширити галузі їх застосування. Аналіз показує, що забезпечити дані вимоги можна при використанні таких принципів побудови :

- змінного складу обладнання, що передбачає наявність обчислювального ядра та змінних спеціалізованих апаратно-програмних модулів, за допомогою яких ядро адаптується до вимог конкретного застосування;
- модульності, який передбачає розробку компонентів нейроорієнтованих комп'ютерних систем у вигляді модулів, що мають вихід на стандартний інтерфейс;
- конвеєризації та просторового паралелізму обробки даних у спеціалізованих апаратно- програмних модулях;
- відкритості програмного забезпечення, що передбачає можливості нарощування та його вдосконалення, максимального використанням стандартних драйверів та програмних засобів;
- узгодженості та адаптації апаратно-програмних спеціалізованих

модулів до інтенсивності надходження даних і структури нейроалгоритмів;

- програмованості архітектури шляхом використання репрограмованих логічних інтегральних мікросхем.

2.3 Алгоритм та структура процесорного елементу для сигмоїдальних функцій активації

Комп'ютерна система може бути презентована у вигляді постійної частини B - універсального обчислювального ядра та змінної частини V - спеціалізованих модулів, що реалізують основні операції нейроалгоритмів. Структура нейроорієнтованої комп'ютерної системи показана на малюнку 3, де ВРР - це багатопортова пам'ять.

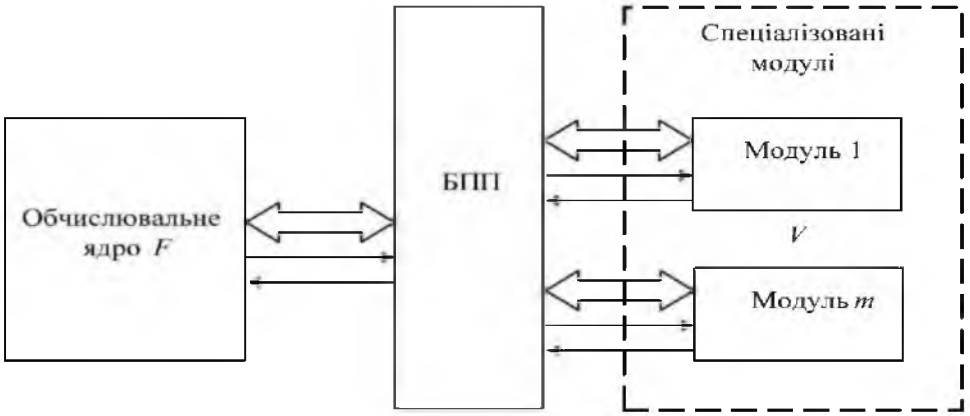


Рисунок 2.1 – Структура нейроорієнтованої комп'ютерної системи

Основними компонентами нейроорієнтованої комп'ютерної системи є ядро комп'ютера, ряд спеціалізованих модулів та ВРР. Паралельність обробки даних у нейроорієнтованій комп'ютерній системі ставить вимоги до організації обміну між ядром комп'ютера та низкою спеціалізованих модулів. Такий обмін у нейроорієнтованій комп'ютерній системі повинен реалізовуватись за допомогою ВРР, що забезпечує паралельний доступ до великої кількості даних як з ядра комп'ютера, так і із спеціалізованих модулів. Нейропроцесор (нейрочіп)

є основою обчислювального ядра нейроорієнтованої комп'ютерної системи. Він містить ряд інструкцій, які добре підходять для виконання основних операцій з нейро-алгоритму, а також повний набір інструкцій загального призначення. Переважна більшість алгоритмів нейропарадигми обмежується виконанням обмеженої кількості основних операцій, таких як «додавання - множення».

Основними перевагами нейрочіпів є:

- відносно вища продуктивність (порівняно з центральним процесором);

- спрощена реалізація з'єднань "всі з усіма" (для розробника нейронних мереж);

- низьке споживання енергії;

- відносно низька ціна.

- Основні недоліки:

- висока структурна складність і низька надійність системи;

- велика складність ефективного здійснення процесу навчання, самонавчання та самоорганізації;

- значне збільшення енергоспоживання та втрата швидкості при одночасному збільшенні ступеня інтеграції нейрочіпів;

Жорстко визначена топологія.

Особливості архітектури та технічні властивості нейропроцесора визначає особливості обчислювального ядра нейроорієнтованої комп'ютерної системи. До таких властивостей входить: довжина інформаційного слова; Кількість основних команд і час їх виконання; Місткість зберігання адрес; Дані в пам'яті та обсяг пам'яті програми та кількість внутрішніх реєстрів.

Структура нейроорієнтованих комп'ютерних систем залежить від конкретних вимог та набору нейро-алгоритмів, що використовуються для вирішення проблем. При вирішенні певної проблеми алгоритм вирішення проблем розподіляється між пристроями та

$$N = N_F + N_V \tag{2.5}$$

де N_F - множина алгоритмів, які виконуються на обладнанні F;

N_V - множина алгоритмів, які виконуються на обладнанні V.

В залежності від співвідношення N_F і N_V комп'ютерні нейроорієнтовані системи діляться на такі типи:

з переважним використанням процесорного ядра (постійного обладнання F), коли

$$N_F \wedge N, N_V \wedge \wedge N_F >> N_V \tag{2.6}$$

з переважним використанням спеціалізованих модулів (змінної частини V), коли

$$N_F \wedge 0, N_V \wedge N, N_F << N_V \tag{2.7}$$

з рівномірним використанням постійного обладнання F і змінної частини V, коли $N_F \sim \mathrm{N_V}$.

Перший тип нейроорієнтованих комп'ютерних систем характеризується тим, що головна кількість обчислювальної потужності зосереджена в ядрі процесора.

У другому типі нейроорієнтованої комп'ютерної системи основні алгоритми обчислень реалізовані за допомогою спеціалізованих модулів, а ядро процесора використовується для виконання допоміжних службових функцій.

Третій тип нейроорієнтованих комп'ютерних систем характеризується тим, що ядро процесора забезпечує алгоритми управління, операцій вводу-виводу та сервісні функції, а також спеціальні модулі - реалізація автоматизованих нейро-алгоритмів, що вимагають великої кількості обчислень.

У більшості випадків для обробки даних нейронної мережі та реалізації алгоритмів видобутку даних необхідно нормалізувати вхідні дані. Нормалізація - це метод попередньої обробки вхідних даних (навчальних, тестових та робочих зразків), при якому значення ознак, що складають вхідний вектор, зменшуються до певного діапазону, зазвичай цього діапазону [0, 1] або [-1 , 1]. Існує багато способів нормалізації вхідних значень. Найпростішою, але найбільш

ефективною в більшості випадків є лінійна нормалізація. Якщо вихідні дані потрібно зменшити до діапазону [0, 1], це робиться наступним чином:

$$x_i^* = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{2.8}$$

Для приведення початкових даних до діапазону [-1, 1] лінійна нормалізація здійснюється таким чиом:

$$x_i^* = \frac{x_i}{x_{\max}} \tag{2.9}$$

Коли вхідні дані щільно заповнюють заданий інтервал, найкраще використовувати лінійну нормалізацію, оскільки вона не вимагає складних обчислень.

Найважливішими даними нормалізації роботи є визначення максимального та мінімального числа з масиву чисел. Щоб забезпечити упаковку потокових даних у реальному часі при використанні таких операторів, потрібна висока швидкість, яка може бути досягнута паралелізацією обчислювального процесу та апаратної реалізації. Тому актуальною проблемою є розробка апаратно-орієнтованих паралельних алгоритмів для визначення максимального та мінімального числа з масиву чисел.

Відомі методи визначення максимального та мінімального числа з масиву чисел засновані на послідовному порівнянні значень кожного числа, починаючи з іншого, з поточними значеннями максимального числа, які на першому кроці приймають значення першого числа. Якщо значення декількох поточних значень є максимальним числом, воно стає поточним максимальним значенням. Отже, кожна ітерація обчислення в поточному значенні максимальної кількості переданих найбільших значень з переданої частини масиву після обчислення поточних значень є максимальним числом у загальній масі [37-38].

Операції з визначення максимального та мінімального чиселю з масиву чисел мають ряд спеціальних особливостей, які не дозволяють використовувати

їх використання за допомогою відомих обчислювальних методів та алгоритмів. Відоме обладнання в основному орієнтованому на реалізацію алгоритмів з перевагою обчислювальних операцій перед логічними, і вони не враховують особливості обчислення максимального та мінімального чисел з масивом. Особливість сучасного обладнання - неефективність використання багатобітових операційних пристроїв, що сприяє зменшенню продуктивності та ефективності обладнання [39-42].

З аналізу літературних джерел можна зробити висновки, що недоліком відомих методів та алгоритмів є те, що вони не орієнтовані на апаратну реалізацію, а використання існуючих апаратних засобів для обчислення максимального та мінімального чисел із масиву чисел є малоефективним.

Тому метою роботи є розроблення алгоритмів і спеціалізованих НВІС-структур для визначення максимальних і мінімальних значень з високою ефективністю використаня обладнання.

Паралельні алгоритми та НВІС-структури пристроїв реалізація сигмоїдальних функцій активації. Аналіз методів і алгоритмів обчислення максимальних і мінімальних значень із масиву чисел показав, що для НВІС-реалізацій найефективнішими є алгоритмами, які грунтуються на методі порозрядного порівняння [42]. Обчислення максимального $A_{\max}$ і мінімального $A_{\min}$ чисел із групи чисел $A_1, A_2, ..., A_j, ...A_m,$ за таким методом 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:

$$\overline{A_{i\max}} = \bigwedge_{j=1}^{m} \overline{a_{ji} \wedge y_{ij}}, y_{1j} = 1, \tag{2.10}$$

$$A_{i\min} = \bigwedge_{j=1}^{m} \overline{a_{ji} \wedge z_{ij}}, z_{1j} = 1, \tag{2.11}$$

де y_{ij}, z_{ij} - i-і розряди j-х слів управління;

a_{ji} - i-розряд j-о числа; m-кількість чисел у групі.

Формування $(i+1)-x$ розрядів j-х слів управління виконується за формулами (2.12) і (2.13) та виконання наступних логічних обчислень :

$$y_{(i+1)j} = (\overline{A}_{i\max} \vee x_{ji}) \wedge y_{ij}, \tag{2.12}$$

$$z_{(i+1)j} = (A_{i\min} \vee x_{ji}) \wedge z_{ij}, \tag{2.13}$$

Процес синтезу паралельних НВІС-структур для обчислення максимальних і мінімальних чисел із групи чисел зводиться до виконання наступних етапів:

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

Аналіз алгоритмів паралельного обчислення максимальних і мінімальних значень із групи чисел на основі методу порозрядного порівняння дозволив виділити для НВІС-реалізації базову операцію. Така базова операція включає формування розрядів слів управління за формулами $y_{(i+1)j} = (\overline{A}_{i\max} \vee x_{ji}) \wedge y_{ij}$ і $z_{(i+1)j} = (A_{i\min} \vee x_{ji}) \wedge z_{ij}$ та виконання наступних логічних обчислень:

$$\overline{A_{ji\max}} = \overline{a_{ji} \wedge y_{ij}}, \tag{2.14}$$

$$\overline{A_{ji\min}} = \overline{\overline{a}_{ji} \wedge z_{ij}}, \tag{2.15}$$

Виділена базова операція реалізується у вигляді ПЕ, схеми яких наведені на рисунку 2.2, де:

а - ПЕ однотактного пристрою;

б - ПЕ конвеєрного пристрою;

в - ПЕ пристрою з вертикальним опрацюванням вхідних чисел.

Особливістю розроблених ПЕ є використання спільних шин результатів, підключення до яких здійснюється за допомогою логічних елементів *2І-НЕ* з відкритим колектором. Вертикальне підключення до спільних шин результатів забезпечує збільшення розміру масиву чисел, з якого визначаються максимальне і мінімальне значення. Використання спільних шин результатів забезпечує високу швидкодії, яка не залежить від кількості чисел у масиві, а залежить тільки від часу затримки на ПЕ.

Вартість НВІС-пристроїв реалізація сигмоїдальних функцій активації в основному залежить від площі кристала, яка визначається як витратами обладнання (кількість транзисторів), так і кількістю зовнішніх виводів, число яких обмежене рівнем технології та розміром кристалу. Орієнтація структур сортування на НВІС-реалізацію вимагає зменшення числа виводів інтерфейсу та кількості з'єднань між ПЕ. Забезпечити ці вимоги можна використанням паралельно-вертикального алгоритму реалізація сигмоїдальних функцій активації, при якому надходження чисел і видача результатів здійснюється розрядними зрізами.

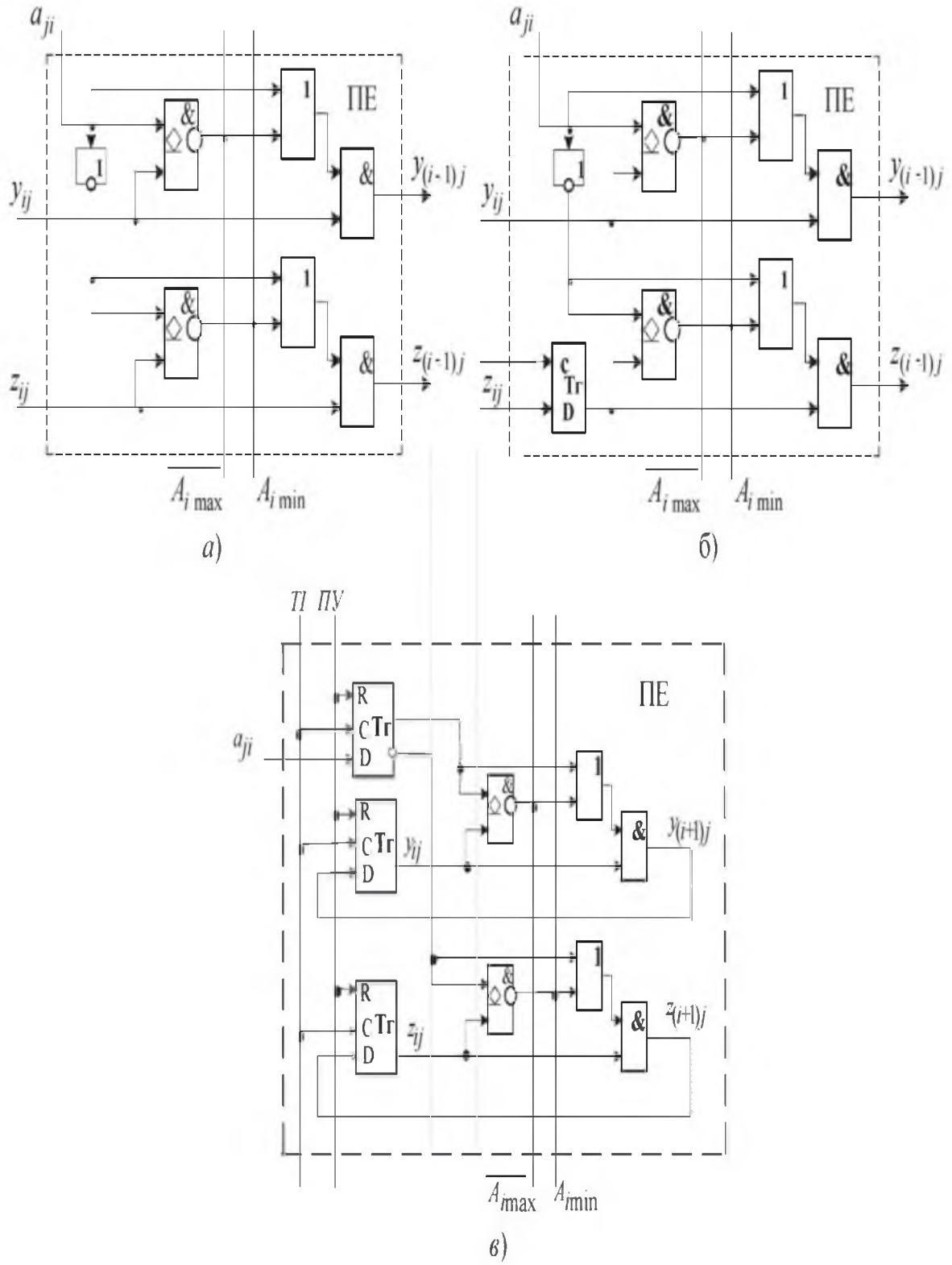


Рисунок 2.2 – де - *а* - ПЕ однотактного пристрою, б - ПЕ конвеєрного пристрою, *в* - ПЕ пристрою з вертикальним опрацюванням вхідних чисел

Структура паралельно-вертикального НВІС-пристрою реалізація сигмоїдальних функцій активації наведена на рисунку 2.3,

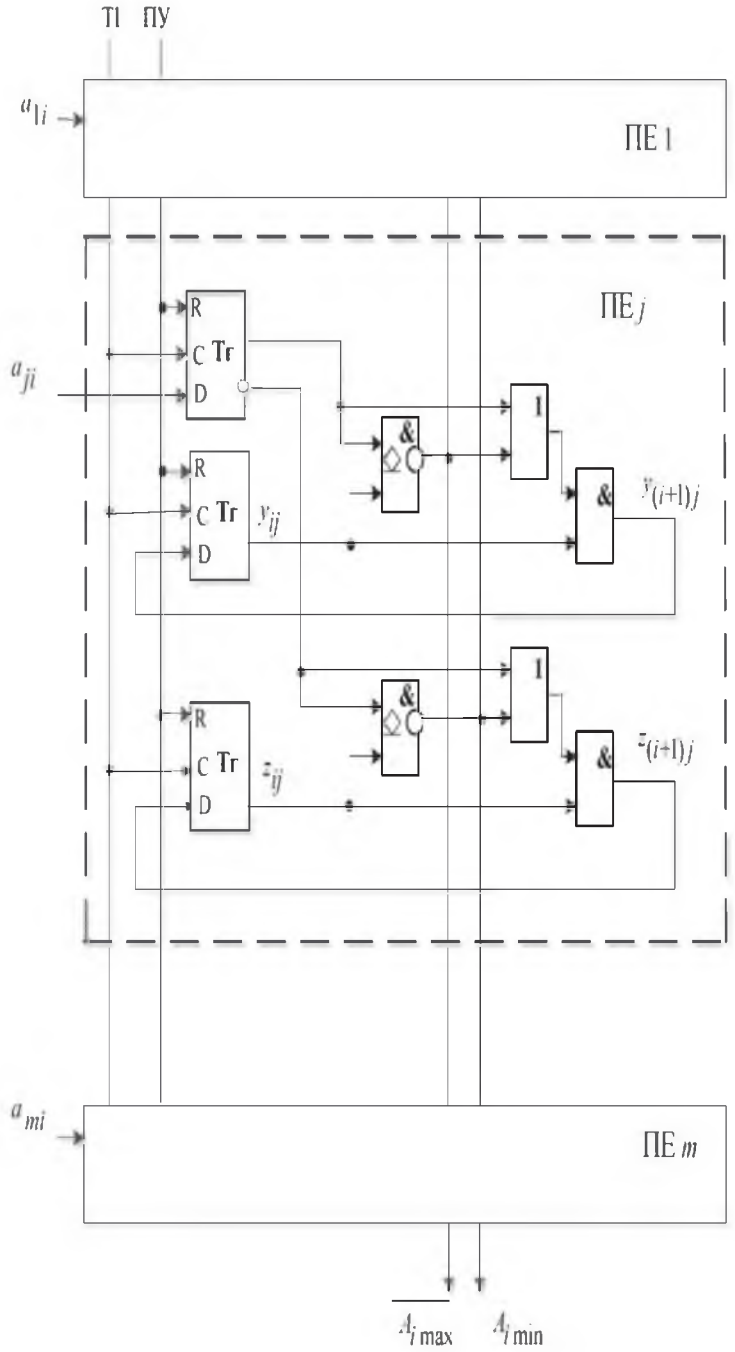


Рисунок 2.3 – Схема паралельно-вертикального НВІС-пристрою обчислення максимальних і мінімаоьних значень

де TI - тактовий вхід;

$ПУ$ - вхід початкової установки тригерів;

$a_1,...,a_m$ - однорозрядні інформаційні входи, де m - кількість чисел, що порівнюються;

$ПЕ_1,...,ПЕ_m$ - $ПЕ$ пристрою з вертикальним опрацюванням вхідних чисел;

Тг1 і Тг2 - D-тригери;

$\overline{A_{\max}}$ і $A_{\min}$ - порозрядний вихід відповідно інверсного максимального та мінімального чисел.

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 Рисунок.2.4, де:

ТІ – тактовий вхід;

У – вхід початкової установки тригерів;

$x_1,\ldots,x_m$ – однорозрядні інформаційні входи, де m – кількість чисел, що порівнюються;

$БП_1,\ldots,БП_m$ – блоки порівняння;

T_1 і T_2 – D-тригери;

Вих – вихід результату.

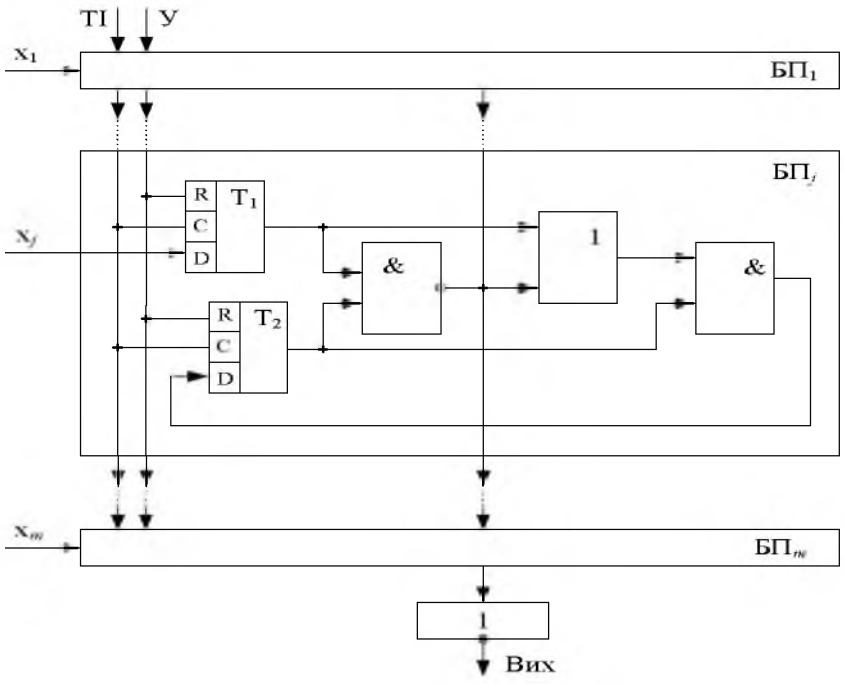


Рисунок. 2.4 – Структура пристрою визначення максимального числа з групи чисел

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 $Т_1$ та $Т_2$ у кожному блоці порівняння $БП_j$ (j=1,…,m) встановлюються у лог.1. Інформація з виходів тригерів $Т_1$ і $Т_2$ (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 $Т_2$ з елемента *І* надходить значення лог.1, а на інформаційний вхід тригера $Т_1$ з однорозрядного інформаційного входу x_j надходить значення старшого розряду j-о числа. Першим тактовим імпульсом у кожному блоці порівняння $БП_j$ у тригер $Т_1$ записується старший розряд j-о числа, а у тригер $Т_2$ 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 $Т_1$ поступає на перший вхід елемента *І-НЕ* з відкритим колектором та перший вхід елемента *АБО*. Лог.1 з виходу тригера $Т_2$ (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 $БП_1$,…,$БП_m$ 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 $БП_1$,…,$БП_m$ на їхніх виходах встановлюється сигнал лог.1, а при лог.0 – інформація з виходів тригерів $Т_1$. Інформація з виходів елементів *АБО* на виходах елементів *І* формує сигнали управління, які надходять на інформаційні входи $Т_2$.

Другим тактовим імпульсом інформація з однорозрядного інформаційного входу x_j (наступний розряд) записується у тригер $Т_1$ кожного блоку порівняння $БП_j$, а у тригер $Т_2$ 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 та мінімального чисел із масиву з m рівний:

$$t_m = (t_{Tг} + 3t_I)n, \tag{2.16}$$

де $t_{Tг}$ - 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:

$$W_m = (3W_{Tг} + 6W_I)m \tag{2.17}$$

де $W_{Tг}$ - затрати обладнання на реалізацію D-тригера;

W_I - затрати обладнання на реалізацію логічних елементів типу *АБО, І, І-НЕ.*

Для опрацювання інтенсивніших потоків даних вимагається збільшення обчислювальної здатності, яке досягається за рахунок збільшення розрядності каналів надходження даних, яка може бути два і більше. Максимальна швидкодія досягається коли розрядність каналів надходження даних дорівнює n, тобто коли одночасно опрацьовуються всі розряди чисел.

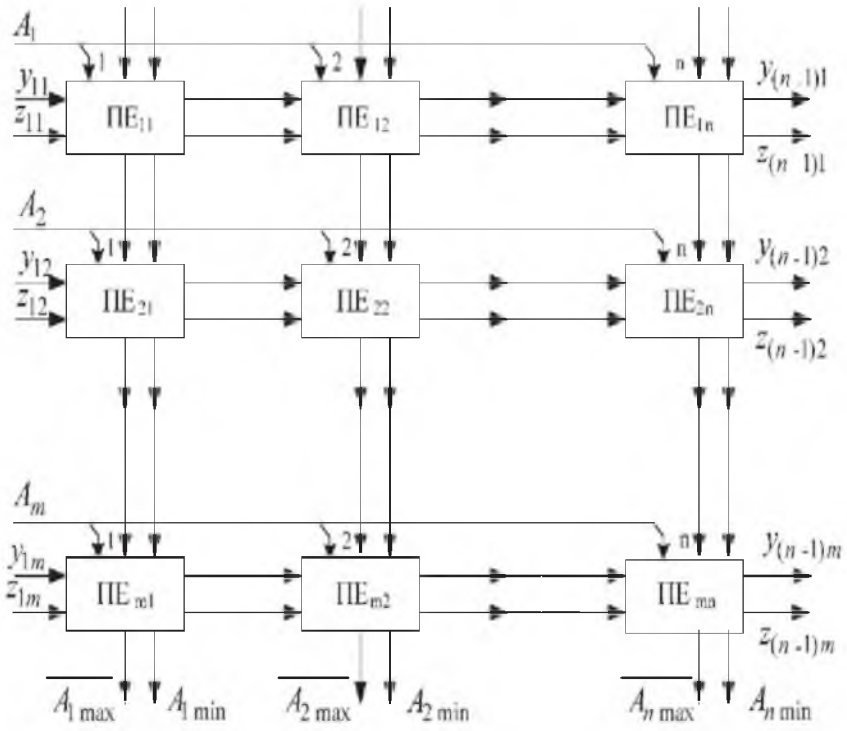


Рисунок 2.5 – Матрична однотактна НВІС-структура пристрою реалізація сигмоїдальних функцій активації

На основі однотактного і конвеєрного ПЕ синтезовано відповідно матричні однотактні (рис.2.3.4) та конвеєрні (рис.2.3.5) паралельні НВІС-структури пристрою обчислення максимальних і мінімальних значень із групи чисел. Кожна із цих структур складається із матриці $(m \times n)$ ПЕ, в якій $ПЕ_{1i}, -ПЕ_{mi}$, i-го стовпчика обчислюють відповідно до формул $\overline{A_{i\max}} = \underset{j=1}{\overset{m}{\wedge}} \overline{a_{ji} \wedge y_{ij}}, y_{1j} = 1,$ і $A_{i\min} = \underset{j=1}{\overset{m}{\wedge}} \overline{\overline{a_{ji}} \wedge z_{ij}}, z_{1j} = 1,$ значення i-х розрядів максимального $\overline{A_{\max}}$ і мінімального $A_{\min}$ чисел та формують у відповідності з формулами $y_{(i+1)j} = (\overline{A}_{i\max} \vee x_{\bar{n}}) \wedge y_{ij},$ і $y_{(i+1)j} = (\overline{A}_{i\max} \vee x_{\bar{n}}) \wedge y_{ij}, z_{(i+1)j} = (A_{i\min} \vee x_{\bar{n}}) \wedge z_{ij},$ значення $(i+1) - x$ слів управління $y(i+1)$ і $z(i+1)$.

Час обчислення максимальних і мінімальних значень у однотактному пристрої визначається за формулою:

$$T_O = 4nt_1, \qquad (2.18)$$

де t_i - час спрацювання логічного елемента "I", n - розрядність чисел.

Апаратні витрати на реалізацію однотактного пристрою дорівнюють:

$$W_O = 7NnW_i, \tag{2.19}$$

де W_i - апаратні витрати на логічні елементи типу І, АБО, І-НЕ.

Особливістю конвеєрної НВІС-структури (рис.2.5) є введення у ПЕ тригерів та використання n блоків пам'яті типу FIFO. Кожний i-ий блок $FIFO_i$,- забезпечує затримку інформації на i тактів. Тривалість конвеєрного такту у такому пристрої визначається за наступною формулою:

$$T_K = t_{Tz} + 4nt_i, \tag{2.20}$$

де t_{Tz} - час спрацювання тригера.

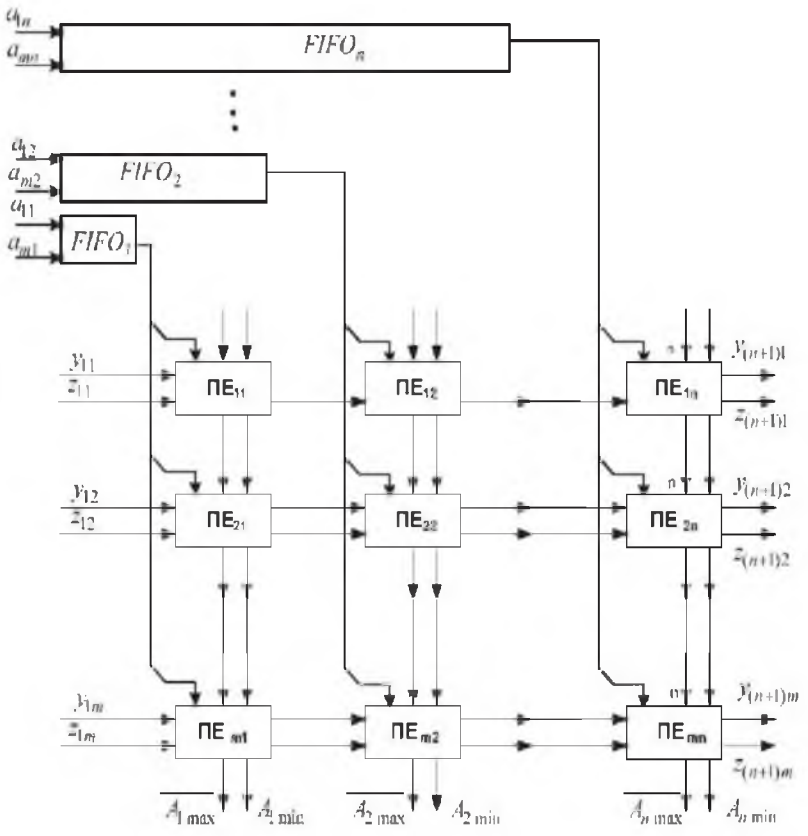


Рисунок 2.6 – Матрична конвеєрна НВІС- реалізація сигмоїдальних функцій активації

Конвеєр, стосовно процесорів, є способом, що використовується при розробці для збільшення інструкційної пропускної здатності (кількості інструкцій, які можуть бути виконані за визначений часовий проміжок). Ідея полягає в тому, щоб розділити обробку комп'ютерної інструкції на послідовність незалежних кроків, зі збереженням результатів у кінці кожного кроку. Це дозволяє управлінським ланцюгам комп'ютера отримувати інструкції зі швидкістю найповільнішого кроку обробки, але таке рішення набагато швидше, ніж виконання всіх цих кроків ексклюзивно для кожної інструкції.

При конвеєрній обробці також використовується суперскалярність – архітектура, що використовує декілька декодерів команд, що можуть навантажувати роботою багато виконавчих блоків. Тобто, якщо в процесі роботи команди, що обробляються конвеєром, не суперечать одна одній і одна не залежить від результату іншої, то такий пристрій може розпаралелити виконання команд.

Апаратні затрати на реалізацію конвеєрного пристрою для обчислення максимальних і мінімальних значень із групи m чисел дорівнюють:

$$W_K = W_{FIFO} + mn(W_{Tг} + 7w_i), \qquad (2.21)$$

де W_{FIFO} і $W_{Tг}$ - апаратні затрати відповідно на пам'ять типу FIFO і тригер.

3 АПАРАТНА РЕАЛІЗАЦІЯ СИГМОЇДАЛЬНИХ ФУНКЦІЙ АКТИВАЦІЇ

3.1 Сигмоїдальні функції активації нейронів

Сигмоїдальні функції активації - найчастіше використовується для нейронних мереж прямого поширення сигналу. Вони є монотонно зростаючими, неперервними і диференційованими. Для опису і аналізу функцій активації сигмоїдальної форми в роботі [1] використано загальний клас функцій:

$$f(x,k,b,T,c) = k + \frac{c}{1+be^{Tx}}, \ \forall x \in R,$$

(3.1)

де $k \in R$, $b \in R^+$, T, $c \in R \setminus \{0\}$, R – множина дійсних чисел (-∞,+∞), R^+ – множина дійсних додатних чисел (0,+∞), R\{0} – множина дійсних чисел за винятком точки 0 (-∞,0) та (0,+∞). Сигмоїдальні нелінійності, які відносяться до цього класу:

- класична сигмоїдальна функція (k=0, c=b=1 та T=−1);

$$f(x) = \frac{1}{1+e^{-x}},$$

(3.2)

- функція, подібна до термодинамічної (k=0, c=b=1 та T=−1/T'):

$$t(x,T') = \frac{1}{1+e^{-x/T'}}, \qquad (3)$$

- гіперболічний тангенс (k=1, c=−2, b=1, T=2):

$$h(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}.$$

(3.4)

Необхідно відмітити, що обчислення сигмоїдальної функції активації (3.2) достатньо виконувати тільки для додатних аргументів x. Для від'ємних значень x її можна знайти з виразу:

$$f(-x) = 1 - f(x). \qquad (3.5)$$

Дійсно, виконавши прості перетворення для виразів (3.2) і (3.4), отримаємо:

$$f(-x)=\frac{1}{1+e^{x}}=\frac{1}{1+\frac{1}{e^{-x}}}=\frac{e^{-x}}{1+e^{-x}}=\frac{1+e^{-x}-1}{1+e^{-x}}=1-f(x),$$

Нелінійну функцію гіперболічного тангенса (4) можна обчислити через класичну сигмоїдальну функцію (2).

$$h(x)=1+\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}-1=\frac{2e^{x}}{e^{x}+e^{-x}}-1=\frac{2}{1+e^{-2x}}-1=2f(2x)-1 \quad , \qquad (3.6)$$

$$h(-x)=2f(-2x)-1=2\left[1-f(2x)\right]-1=1-2f(2x). \qquad (3.7)$$

Для оцінки точності апроксимації використовується максимальна і середня похибки [2]. Середня абсолютна ε_{ave}, максимальна абсолютна $\varepsilon_{\max}$ похибки функції $f(x)$, яка апроксимується функцією $\overline{f}(x)$ в інтервалі $(x_{\min}, x_{\max})$ визначаються як:

$$\varepsilon_{ave}=\frac{\sum_{i=0}^{N_p-1}\left|\overline{f}(x_i)-f(x_i)\right|}{N_p}, \qquad (3.8)$$

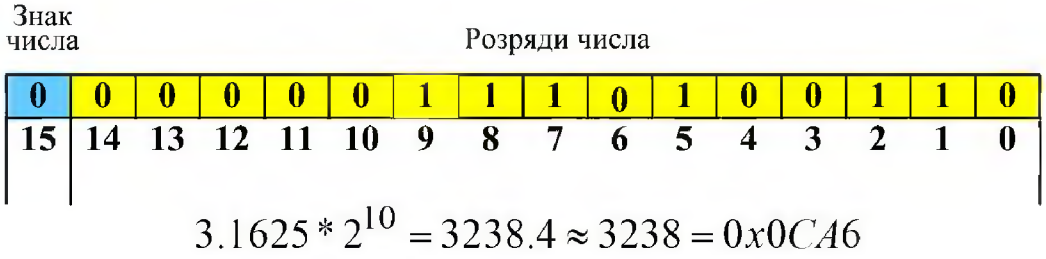
$$\varepsilon_{\max}=\max\left|\overline{f}(x_i)-f(x_i)\right|,\ i=0,\ldots,N_p, \qquad (3.9)$$

де N_p - кількість точок, на які розбивається інтервал $(x_{\min}, x_{\max})$.

Ефективність апроксимації порівнюється по досягнутій точності, швидкості та апаратним ресурсам.

Арифметичні операції в нейронах виконуються над дійсними числами. При реалізації арифметичних операцій на FPGA над дійсними числами з фіксованою та плаваючою крапками зростає час їх виконання та апаратні ресурси необхідні для їх реалізації. Тому при реалізації компонент штучних нейронів на мові VHDL дійсні числа перетворювалися до цілих шляхом їх домноження на 2^{10} і відсіканням дробової частини. Формат представлення дійсних чисел має розмірність 16 біт: 1 біт – знаковий і 15 біт – для зберігання отриманого цілого числа. Максимальна ціла частина дробового числа не повинна перевищувати 2^4-1=15. Необхідно зауважити, що вхідні дані в нейронних мережах нормуються. Представлення дійсного числа 3.1625 у форматі цілих чисел зображеного на рис. 1. Для цього це число домножуємо на 2^{10} = 1024 і відсікаємо дробову частину:

3.1625*1024=3238.4≈3238. В шістнадцятковій системі числення це число рівне: 3238_{10}=0x0CA6.



$$3.1625 * 2^{10} = 3238.4 \approx 3238 = 0x0CA6$$

Рисунок 3.1 - Представлення дійсного числа 3.1625 у форматі знакового цілого числа

3.2 Методи апроксимації сигмоїдальних функцій

Розглянемо декілька алгоритмів PWL, які відрізняються кількістю точок, розміщенням початкової і кінцевої точок на апроксимуючих лініях та критеріями їх вибору.

1. Piecewise linear approximation of a nonlinear function (PLAN approximation) [3.2, 3.3]. В цьому методі апроксимація виконується виразом [3.2]:

$$f(x) = \begin{cases} 1, & |x| \geq 5.0 \\ 0.03125 * |x| + 0.84375, & 2.375 \leq |x| < 5.0 \\ 0.125 * |x| + 0.625, & 1.0 \leq |x| < 2.375 \\ 0.25 * |x| + 0.5, & 0 \leq |x| < 1.0 \end{cases} \quad (3.10)$$

Обчислення повинні виконуватися тільки для вхідних даних x по модулю. Для від'ємних вхідних даних x сигмоїдальна функція обчислюється виразом (3.5). Перетворимо вираз (3.10) в цілочисельний, помноживши його доданки без змінної $|x|$ на 2^{10} :

$$f(x)=\begin{cases}1024, & |x|\geq 5120\\ 2^{-5}*|x|+864, & 2432\leq|x|<5120\\ 2^{-3}*|x|+640, & 1024\leq|x|<2432\\ 2^{-2}*|x|+512, & 0\leq|x|<1024\end{cases}. \tag{3.11}$$

Для реалізації сигмоїдальної функції відповідно до виразу (3.11) розроблена структура пристрою, яка наведена на рисунку 3.2, де Рг – регістр, СП схема порівняння, См –суматор.

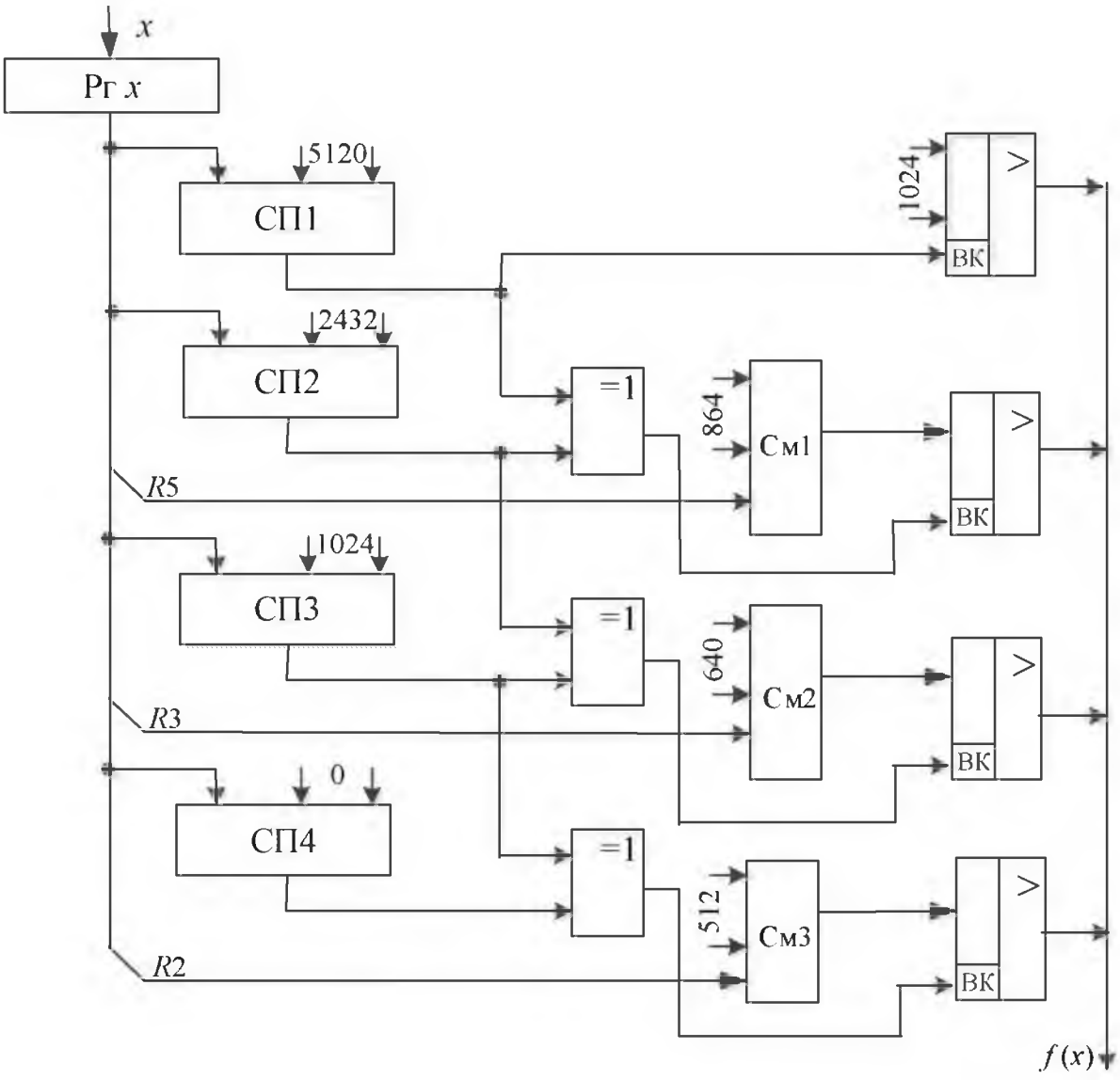


Рисунок 3.2 - Структура пристрою обчислення сигмоїдальної функції відповідно до виразу (11)

Обчислення сигмоїдальної функції в даному пристрої виконується тільки з використанням трьох суматорів См та чотирьох сем порівняння СП. Множення на 2^{-2}, 2^{-3} і 2^{-5} здійснюється шляхом відповідного з'єднання. Вибір результатів обчислення сигмоїдальної функції, які отримуються на виходах См1, См2 і См3, здійснюється за результатами порівняння вхідних даних x з числами 5120, 2432, 1024 та 0. На перші входи СП поступають вхідні дані, які порівнюються з даними (B_j), які установлені на других входах. У залежності від вхідних даних x на виходах СП формується результат порівняння

$$ВихСП = \begin{cases} 0, & коли \quad x < B_j \\ 1, & коли \quad x \geq B_j \end{cases}. \tag{3.12}$$

Інформація з виходу СП надходить на входи логічних елементів виключне АБО, які формують вихідний сигнал у відповідності з виразом:

$$ВикАБО = \begin{cases} 0, & коли \quad 1Bx = 2Bx \\ 1, & коли \quad 1Bx \neq 2Bx \end{cases}. \tag{3.13}$$

Сигнали з СП1 та виходів логічних елементів виключне АБО поступають на входи ВК шинних формувачів, які працюють так: на ВК сигнал лог. 0 -вихід у третьому; на ВК сигнал ог.1 - передача даних.

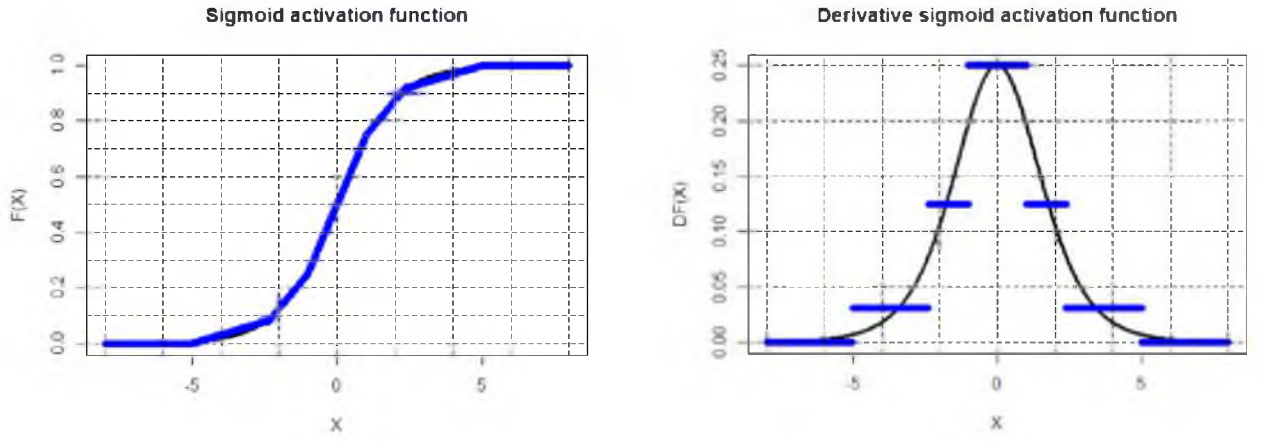
Час обчислення сигмоїдальної функції визначається за формулою:

$$t_1 = t_{Рг} + t_{СП} + t_{ВикАБО} + t_{ШФ}, \tag{3.14}$$

де $t_{Рг}$, $t_{СП}$, $t_{ВикАБО}$, $t_{ШФ}$ – час спрацювання відповідно регістра, схеми порівняння, логічного елемента виключне АБО та шинного формувача.

3.3. Моделювання сигмоїдальної функцій активації

Моделювання сигмоїдальної функції та її PLAN апроксимації зображено на рисунку 3.3а, а їх похідних на рисунку 3.3б.



а) б)

Рисунок 3.3 - Вигляд сигмоїдальної функції

а) PLAN апроксимації; б) –похідних та PLAN апроксимації

Розіб'ємо діапазон (-8, 8) на $N_p = 1000$ точок. В цьому діапазоні середня і максимальна похибки PLAN апроксимації сигмоїдальної функції рівні $\varepsilon_{ave} =$ 0.00587, $\varepsilon_{\max} = 0.0185$, та її похідної $d\varepsilon_{ave} = 0.01412$, $d\varepsilon_{\max} = 0.07088$.

Вигляд абсолютної похибки між сигмоїдальною функцією та її PLAN апроксимацією (крива чорного кольору), між похідними сигмоїдальної функції і її PLAN апроксимацією (крива синього кольору) приведений на рисунку 3.4.

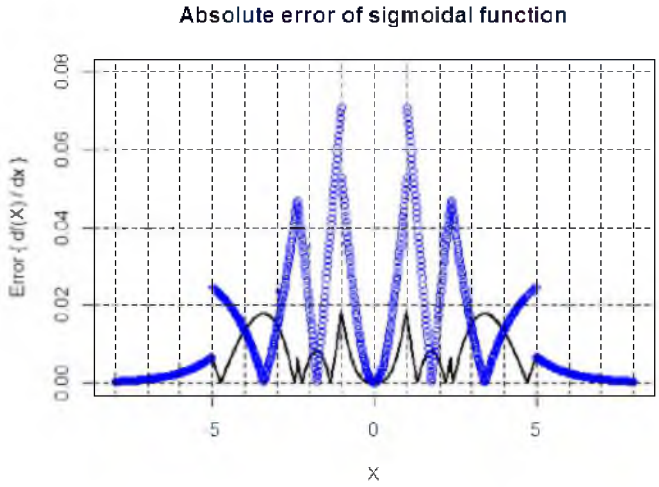


Рисунок 3. 4 - Вигляд абсолютної похибки похідних сигмоїдальної функції та її PLAN апроксимації

Апроксимація сигмоїдальної функції кривою другого порядку. Для вхідних даних з діапазону $(0, x_{\max})$ сигмоїдальна функція апроксимується поліномом [3.2, 3.3]:

$$\overline{f}(x) = c + bx + ax^2. \tag{3.15}$$

В рівнянні (15) невідомими є коефіцієнти *a*, *b*, *c*, які визначаються методом найменших квадратів. Система рівнянь для їх обчислення має вигляд:

$$\begin{bmatrix} \sum_{i=0}^{N_p-1} x_i^2 & \sum_{i=0}^{N_p-1} x_i^3 & \sum_{i=0}^{N_p-1} x_i^4 \\ \sum_{i=0}^{N_p-1} x_i & \sum_{i=0}^{N_p-1} x_i^2 & \sum_{i=0}^{N_p-1} x_i^3 \\ N_p & \sum_{i=0}^{N_p-1} x_i & \sum_{i=0}^{N_p-1} x_i^2 \end{bmatrix} * \begin{bmatrix} c \\ b \\ a \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{N_p-1} f(x_i) x_i^2 \\ \sum_{i=0}^{N_p-1} f(x_i) x_i \\ \sum_{i=0}^{N_p-1} f(x_i) \end{bmatrix}. \tag{3.16}$$

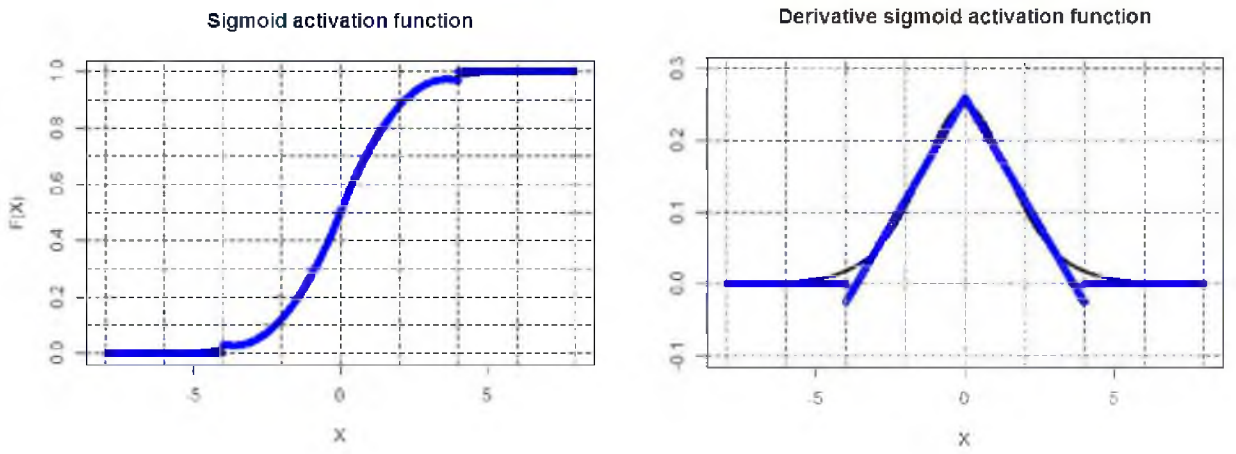
Розбиваємо інтервал $(0,4)$ на $N_p = 1000$ точок. Використовуючи значення x_i та $f(x_i)$ в цих точках, формуємо систему рівнянь (16). Розв'язавши її отримаємо коефіцієнти *a*, *b*, *c*. Вираз для апроксимації сигмоїдальної функції:

$$f(x) = \begin{cases} 1, & x \geq 4.0 \\ -0.03577 * x^2 + 0.25908 * x + 0.5038, & 0 \leq x < 4.0 \end{cases}. \tag{3.17}$$

Цілочисельний вираз (17) має вигляд:

$$f(x) = \begin{cases} 1024, & x \geq 4096 \\ -36 * 2^{-20} * x^2 + 265 * 2^{-10} * x + 515, & 0 \leq x < 4096 \end{cases}. \tag{3.18}$$

В діапазоні (-8, 8) середня і максимальна похибки апроксимації сигмоїдальної функції поліномом другого порядку рівні $\varepsilon_{ave} = 0.00426$, $\varepsilon_{\max} = 0.01798$, та її похідної $d\varepsilon_{ave} = 0.00769$, $d\varepsilon_{\max} = 0.04388$. Сигмоїдальна функція та апроксимація її поліномом другого порядку зображено на рисунку 3.5,а, а їх похідних на рисунку3.5,б. Чорним кольором на рисунку 3.5 зображена сигмоїдальна функція і її похідна, а синім кольором – їх апроксимації.



а) б)

Рисунок 3.5 - Вигляд сигмоїдальної функції

а) апроксимація поліномом другого порядку; б) похідних та апроксимації

На рисунку 3.6 зображено абсолютні похибки між сигмоїдальною функцією та її апроксимацією поліномом другого порядку (чорний колір) і між похідною сигмоїдальної функції та похідною її апроксимації (синій колір).

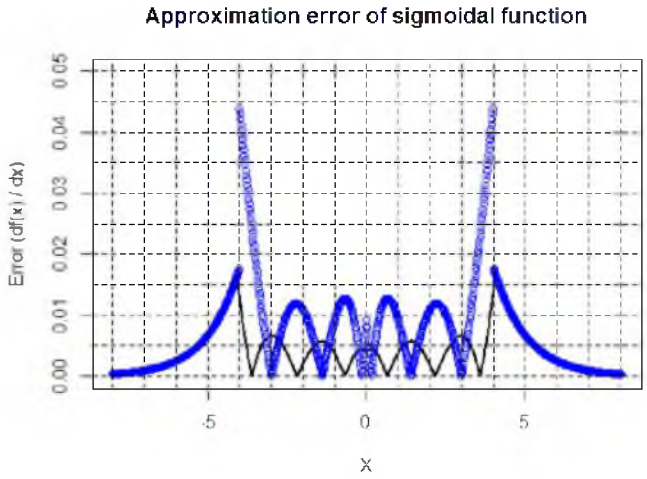


Рисунок 3.6 - Вигляд абсолютної похибки похідних сигмоїдальної функції та її апроксимації поліномом другого порядку

Для реалізації сигмоїдальної функції (3.15) розроблена структура пристрою, яка наведена на рисунку 3.7, де БМ – блок множення.

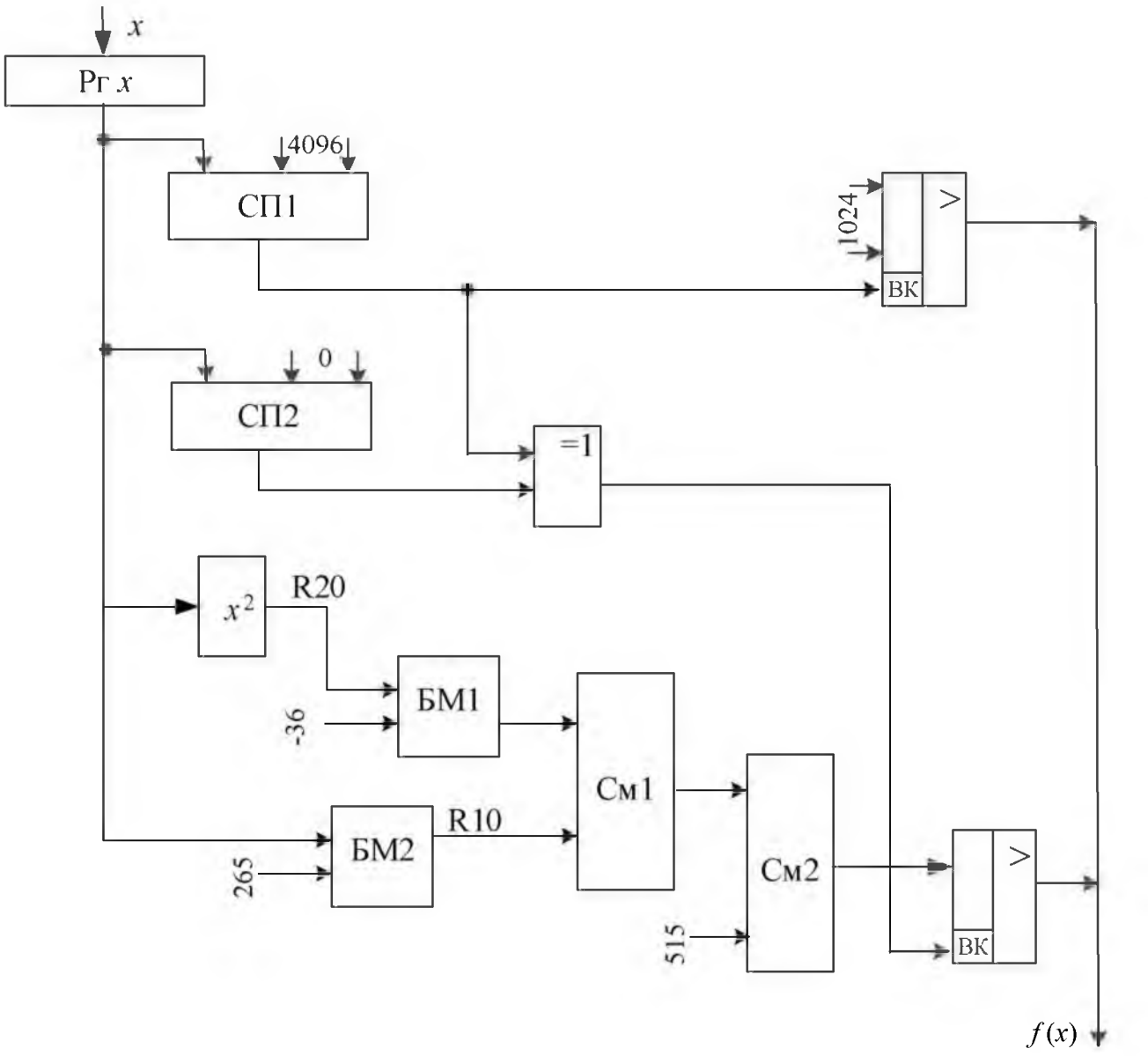


Рисунок 3.7 - Структура пристрою реалізації сигмоїдальної функції за виразом

У даному пристрої для реалізації сигмоїдальної функції використовуються два блоки множення та блок піднесення до квадрату, що збільшує затрати на його реалізацію. Час обчислення сигмоїдальної функції в даному пристрої визначається за формулою:

$$t_2 = t_{Рг} + 2t_{БМ} + t_{БПК} + 2t_{См} + t_{ШФ}, \qquad (3.19)$$

де $t_{БМ}$, $t_{БПК}$, $t_{См}$ – час спрацювання відповідно блоку множення, блоку піднесення до квадрату та суматора.

В роботі [2, 3] для апроксимації сигмоїдальної функції поліномом другого порядку використовується спрощений вираз:

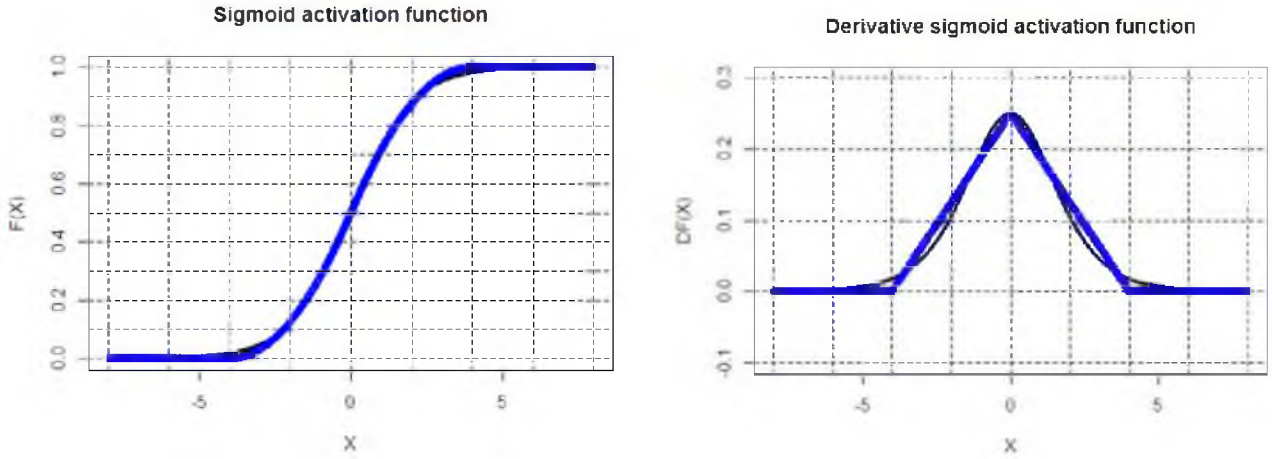
$$f(x)=\begin{cases}1, & x\geq 4.0\\ -0.03125*x^2+0.25*x+0.5, & 0\leq x<4.0\end{cases}, \qquad (3.20)$$

реалізація якого вимагає тільки одного перемножувача, регістрів зсуву та суматорів.

В цілочисельному форматі вираз (20) має вигляд:

$$f(x)=\begin{cases}1024, & x\geq 4096\\ -2^{-15}*x^2+2^{-2}*x+512, & 0\leq x<4096\end{cases}, \qquad (3.21)$$

Сигмоїдальна функція і апроксимація її виразом (3.21) в діапазоні (-8, 8) зображені на рисунку 3.8а, а їх похідні – на рисунку 3.8б. На цих рисунках чорним кольором зображена сигмоїдальна функція та її похідна, а синім – апроксимація сигмоїдальної функції та її похідної.



а) б)

Рисунок 3.8. Графіки а) – сигмоїдальної функції та її апроксимації;
б) –похідних сигмоїдальної функції та її апроксимації

Середня і максимальна похибки апроксимації сигмоїдальної функції виразом (3.16) в діапазоні (-8, 8) рівні $\varepsilon_{ave}=0.00774$, $\varepsilon_{\max}=0.02160$, та їх похідних $d\varepsilon_{ave}=$ 0.00877, $d\varepsilon_{\max}=$ 0.02375. Вигляд абсолютних похибок зображений на рисунку 3.9 (чорний колір – похибка між сигмоїдальною функцією та апроксимуючим виразом (3.16), синій колір – похибка між їх похідними).

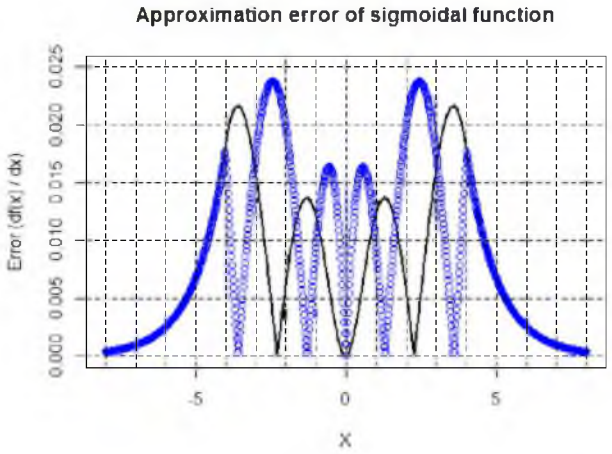


Рисунок 3. 9 - Вигляд абсолютних похибок між сигмоїдальною функцією та її апроксимацією виразом (3.16) та між їх похідними

Для реалізації сигмоїдальної функції (3.21) розроблена структура пристрою, яка наведена на рисунку 3.10, де Вд – віднімач.

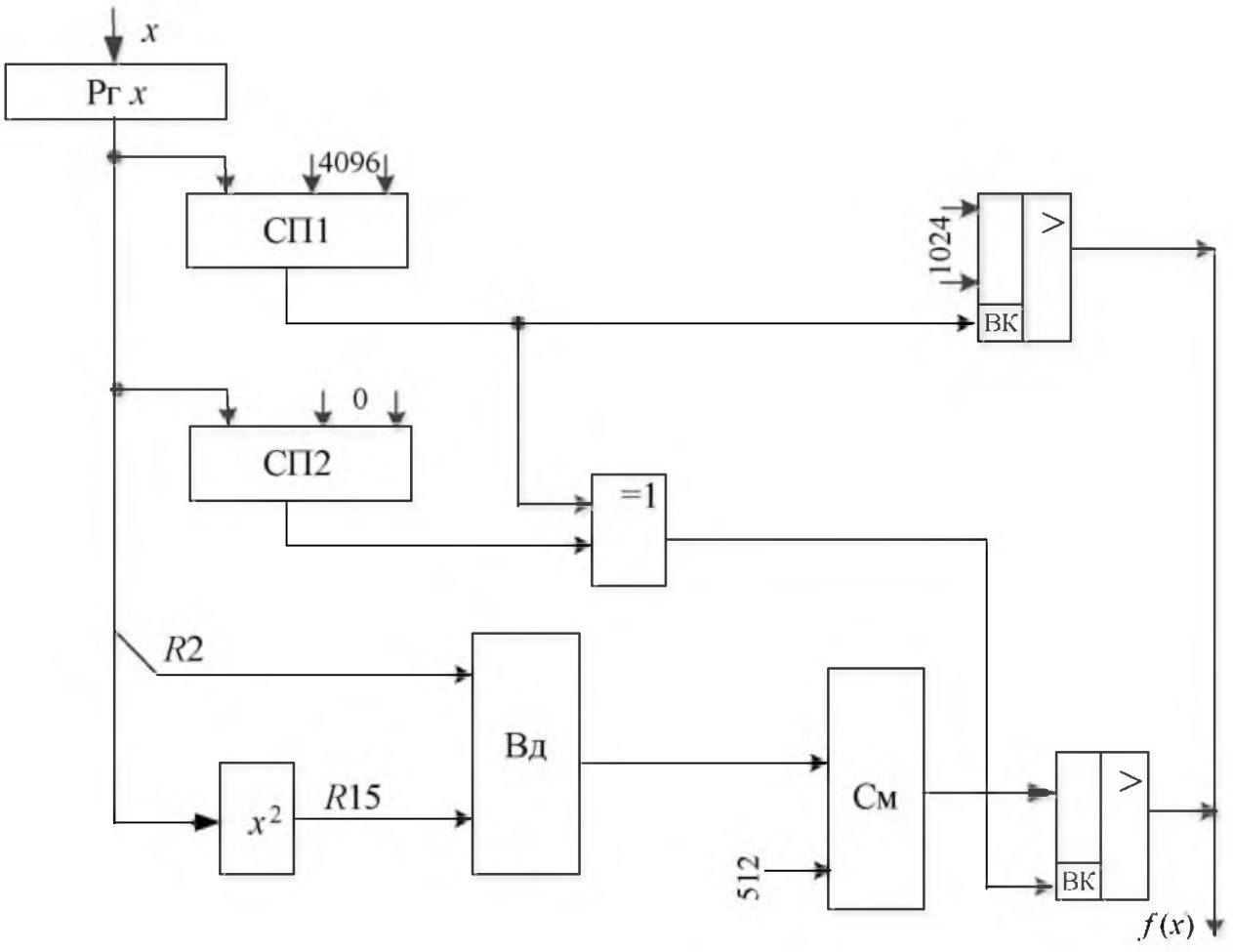


Рисунок 3.10 - Структура пристрою реалізації сигмоїдальної функції за виразом 21

У даному пристрої для реалізації сигмоїдальної функції застосовується блок піднесення до квадрату, віднімач та суматор. У порівнянні з пристроєм рис.7, даний пристрій вимагає для своєї реалізації менших затрат обладнання. Час обчислення сигмоїдальної функції в даному пристрої визначається за формулою:

$$t_З = t_{Рг} + t_{БПК} + t_{См} + t_{Вд} + t_{ШФ}, \quad (3.19)$$

де $t_{Вд}$ – час віднімання.

3.4. Реалізація сигмоїдальних функцій на FPGA

Реалізація сигмоїдальних функцій виконувалася в середовищі розробки Quartus II для FPGA EP3C16F484C6 сімейства Cyclone III на мові програмування апаратури VHDL.

Для цього використовувалися методи апроксимації сигмоїдальної функції (вирази (3.11), (3.18), (3.21). Кожний з цих виразів реалізований у вигляді символу. На рисунку 3.11 зображено символ для PLAN апроксимації сигмоїдальної функції (вираз 3.11). Інші два символи (FA_Sigm_N2, FA_Sigm_N3) мають аналогічні входи та виходи. Входами цих символів є: Clk – вхід синхронізації, IN[15..0] – сума зважених входів нейрона розрядністю 16, а виходом – Out[15..0] – значення апроксимації функції активації розрядністю 16. Сигмоїдальна функція обчислюється по передньому фронту імпульсів, які поступають на вхід Clk.

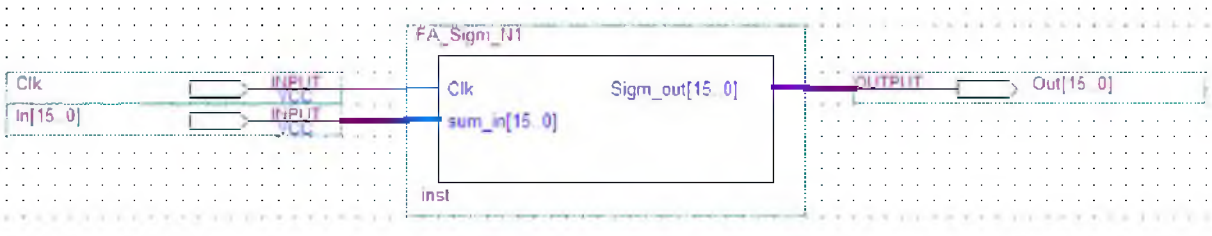


Рисунок 3.11 - Зовнішній вигляд символу FA_Sigm_N1

На рисунку 3.12 наведений фрагмент часової діаграми роботи модуля, який обчислює сигмоїдальну функцію, використовуючи PLAN апроксимацію. Часова

діаграма отримана, використовуючи моделювання в часовій області (timing modulation) в середовищі розробки Quartus II.

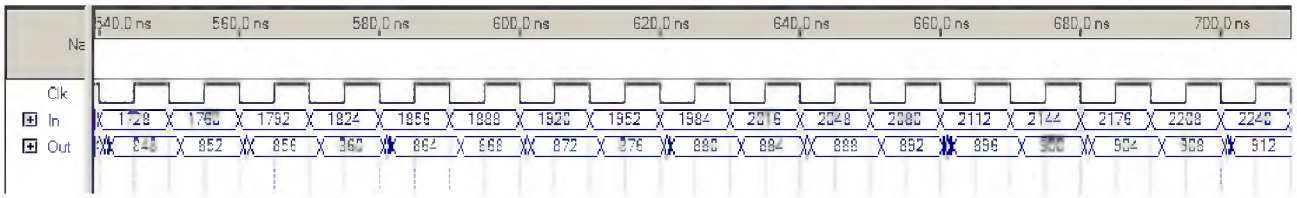


Рисунок 3.12 - Часові діаграми роботи модуля сигмоїдальної функції FA_Sigm_N1

На входи модуля FA_Sigm_N1 подається послідовність імпульсів синхронізації з періодом 10 ns та послідовність зважених сум від 0 до 8192 з кроком 32. На виході суматора отримуємо величину апроксимованої сигмоїдальної функції. На рисунку 3.9 зображений фрагмент з вхідними величинами в діапазоні від 1728 до 2240. Так значенню вхідного сигналу In=1728 відповідає величина сигналу Out=2^{-3}*1728+640=856. Затримка вихідного сигналу по відношенню до імпульсів синхронізації (t_{CO}) для модулів FA_Sigm_N1 і FA_Sigm_N3 складає 27…28 ns, а для модуля FA_Sigm_N3 – 37 ns.

Проведено порівняння по точності апроксимації сигмоїдальної функції виразами (3.11), (3.18) і (3.21) та апаратних ресурсах FPGA, затрачених на їх реалізацію. Апаратні ресурси порівнюються по кількості логічних елементів (ЛЕ) та виводах FPGA EP3C16F484C6 сімейства Cyclone III фірми Altera. Максимальна кількість ЛЕ для цієї FPGA [4] складає 15408, а кількість виводів – 347. Результати порівняння наведені в таблиці 3.1.

Таблиця 3.1 - Порівняння реалізацій апроксимацій сигмоїдальної функції

Символ	ε_{ave}	$\varepsilon_{\max}$	$d\varepsilon_{ave}$	$d\varepsilon_{\max}$	Number LE	Pins
FA_Sigm_N1	0.00587	0.01850	0.01412	0.07088	246/15408	33/347
FA_Sigm_N2	0.00426	0.01798	0.00769	0.04388	368/15408	33/347
FA_Sigm_N3	0.00774	0.02160	0.00877	0.02375	167/15408	33/347

Крім моделювання сигмоїдальних функцій виконано їх реалізацію на стенді DE0 [5]. Значення зваженої суми, що подається на вхід модулів FA_Sigm_N1, FA_Sigm_N2, FA_Sigm_N3 задається з допомогою перемикачів, а величина функції активації виводиться на семисегментний індикатор стенду.

ВИСНОВКИ

1. Обчислення сигмоїдальної функції активації з використанням ПЛІС можна забезпечити при комплексному підході, який охоплює: дослідження методів і алгоритмів . обчислення сигмоїдальної функції активації.

2. Розроблення високоефективних сигмоїдальних функції активації методом порозрядного порівняння найправильніше здійснювати при інтегрованому підході, який охоплює методи, алгоритми, структури і НВІС-технологію та бере до уваги особливості конкретного застосування.

3. Для апроксимації сигмоїдальної функції активації використано методи кусково-лінійної апроксимації (модуль FA_Sigm_N1) та апроксимацію поліномом другого порядку (модулі FA_Sigm_N2, FA_Sigm_N3). В першому методі для перемноження використовується зсуви. В другому методі використовується три перемножувачі, а в третьому тільки один перемножувач та зсуви. Виконано порівняння по точності реалізацій трьох сигмоїдальних функцій. Цифрова апаратна реалізація сигмоїдальної функції активації виконувалася мовою VHDL в середовищі розробки Quartus II. Модулі сигмоїдальних функцій реалізовані на FPGA EP3C16F484C6 сімейства Cyclone III. Виконано моделювання їх роботи в часовій області та порівняння по необхідних апаратних ресурсах і швидкодії. Реалізовані модулі сигмоїдальних функцій будуть використані при проектуванні багатошарових нейронних мереж прямого поширення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Данило А.М., Середа Ю. І. Алгоритми та апаратна реалізація функцій сигмоїдальних функцій активації III Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 26 листопада 2020, Тернопіль, Україна. Тернопіль: ЗУНУ, 2020. с.62

2. Данило А.М., Середа Ю. І. Методи синтезу та створення нейромереж реального часу паралельно-вертикального типу III Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 26 листопада 2020, Тернопіль, Україна. Тернопіль: ЗУНУ, 2020. с.66.

3. Нейроподібні методи, алгоритми та структури обробки сигналів і зображень у реальному часі: монографія / Ю.М. Рашкевич, Р.О. Ткаченко, І.Г Цмоць, Д.Д. Пелешко. Львів: Видавництво Львівської політехніки, 2014. -256 с.

4. Проблемно-ориентированные высокопроизводительные вычислительные системы: В.Ф. Гузик, В.Е. Золотовский: Учебное пособие. Таганрог:Изд-во ТРТУ, 1998. 236 с.

5. Уоссермен Ф.Нейрокомпьютерная техника. – М.: Мир,1992. – 259с.

6. А.В. Палагин, В.Н. Опанасенко. Реконфигуриремые вычислительные системы. – К.: Просвіта, 2006.- 280с.

7. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая Линия-Телеком, 2002. – 382 с.

8. Николаев А.Б., Фоминых И.Б. Нейросетевые методы анализа и обработки данных. Учебное пособие. - М.: МАДИ (ГТУ), 2003, 95с.

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

10. Грибачев В. П. Элементная база аппаратных реализаций нейронных сетей // Компоненты и технологии. 2006. № 8

11. Круг П.Г. Нейронные сети и нейрокомпьютеры: Учебное пособие по курсу «Микропроцессоры». М.: Издательство МЭИ, 2002. 176 с.

12. Проблемы построения и обучения нейронных сетей/ под ред. А.И.Галушкина и В.А.Шахнова. - М. Изд-во Машиностроение. Библиотечка журнала Информационные технологии №1. 1999. 105 с.

13. А.И.Галушкин Некоторые исторические аспекты развития элементной базы вычислительных систем с массовым параллелизмом(80-и90-годы) // Нейрокомпьютер, №1. 2000. - С.68-82

14. С.И.Аряшев, С.Г.Бобков, Е.А.Сидоров Параллельный перепрограммируемый вычислитель для систем обработки информационных сигналов// "Нейроинформатика-99". - Москав, МИФИ. Часть2. С.25-33.

15. Э.Ю. Кирсанов Цифровые нейрокомпьютеры: Архитектура и схемотехника / Под ред. А.И.Галушкина. - Казань: Казанский Гос. У-т. 1995. 131 с.

16. А.И. Власов. Аппаратная реализация нейровычислительных управляющих систем//Приборы и системы управления- 1999, №2, С.61-65.

17. Борисов В.Л., Капитанов В.Д. Методика быстрого создания нейроускорителей// Нейрокопьютеры: разработка и приенение, №1, 2000 год.-С.12-24.

18. Роберт Хехт-Нильсен Нейрокомпьютинг: история, состояние, перспективы// Открытые системы. N4. 1998.

19. А.И. Власов Нейросетевая реализация микропроцессорных систем активной акусто- и виброзащиты// Нейрокомпьютеры:разработка и применение, №1, 2000. С.40-44.

20. M. Petryk, I. Boyko, M. Petryk, J. Fraissard, I. Mudryk Modeling of adsorption and desorption of hydrocarbons in nanoporous catalytic zeolite media using nonlinear Langmuir isotherm. Fourteenth International Conference on Correlation Optics, Chernivtsi, Ukraine, 2019, p. 42.

21. Елементна база нейрообчислювачів-http://opticstoday.com/katalogstatej/stati-na-ukrainskom/nejrokomputeri/elementna-baza-nejroobchislyuvachiv.html

22. Сучасні напрямки розвитку нейрокомп'ютерних технологій-http://www.victoria.lviv.ua/html/oio/html/theme9.htm

23. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы М.: Изд. Дом«Вильямс», 2001. 384с.

25. Проценко В.С. Техніка програмування мовою Сі: Навчальний посібник К.: Либідь, 1993. 224 с.

26. Шилдт Г. Теория и практика С++ СПб.: BHV Санкт-Питербург, 1996. 416 с.

27. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: «Мир», 1979. - 536с.

28. Вирт Н. Алгоритмы + структуры данных = программы. - М.: "Мир", 1985. - 544 с.

29. Гудман С. Хидетниеми С. Введение в разработку и анализ алгоритмов. - М.: "Мир", 1981. - 366 с.

30. Кнут Д. Искусство программирования для ЭВМ. т.3. Сортировка и поиск. М.:Мир, 1976. - 678 с.

31. Мейер Б., Бодуэн К. Методы программирования: В 2-х томах М.: Мир, 1982. 356+368с.

32. Керниган, Б.,Мова програмування Сі. Завдання по мові Сі/Б.Керниган, Д.Ритчи. —М.:ФиС, 1985. — 280 с.

33. Страустрап, Б. Мова програмування Сі ++ /Б.Страуструп. — М.:Радіо та зв'язок, 1991.—352 с.

34. Белецкий, Я. Енциклопедія мови Сі /Я.Белецкий. — М.:Мир, 1992. — 687 с.

35. Белецкий, Я.ТурбоСі++: Нова розробка: навч. посібник для студентів вищих навчальних закладів / Я.Белецкий. — М.'.Машинобудівництво, 1994. — 400с.

36. Пильщиков, В. Н. Збірник вправ по мові Паскаль: навч. Посібник для втузов /В. Н.Пильщиков. — М.:Висш. шк.,1990. —223 с.

37. Кармен, Томас Х., Лейзерон, Чарльз И., Ривест, Рональд Л., Штайн, Клиффорд. Алгоритми: построение и анализ, 2-е издание. :Пер. с англ. - М.: Издательский дом "Вильямс", 2005. - 1296 с.

38. *Кнут Д.* Искусство программирования для ЗВМ: Сортировка и поиск. М., - 1978. - 844с.

39. *Кухарев Г.А.* и др. Техника параллельной обработки бинарних данньїх на СБИС. - М.: Виш. Шк., 1991. - 226 с

40. Пат. № 66138, Україна, МПК 006Б 7/38. Пристрій для обчислення сум парних добутків: Патент на корисну модель / *І.Г. Цмоць, О.В. Скорохода*; заявник і патентовласник Національний університет «Львівська політехніка». - № и201106811; заявл. 30.05.2011; опубл. 26.12.2011, Бюл. № 24. - 8 с..

41. Патент України на винахід №29700. Пристрій для визначення максимального числа з групи чисел. Бюл. №6-11. - 2000. *Рашкевич Ю.М., Зербіно Д.Д, Цмоць І.Г.*

42. *Кун С.* Матричние процессорина СБИС. - М.: Мир, 1991. 672

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

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

45. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика М.: Горячая Линия-Телеком, 2002 382 с.

46. Рутковская Д., Пилиньский Л., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы / Пер. с польского М.: Горячая линия-Телеком, 2007. 452 с.

47. с. Рассел С., Норвиг П. Искусственный интеллект: современный подход / Пер. с английского М.: Вильямс, 2007. 1408 с.

48. Рассел С., Норвиг П. Искусственный интеллект: современный подход / Пер. с английского М.: Вильямс, 2007. 1408 с.

49. Цмоць І.Г. Принципи розробки і оцінка основних характеристик високопродуктивних процесорів на надвеликих інтегральних схемах/ Вісник ДУ “Львівська політехніка”, №349, Львів, 1998 - с.5-11.

50. Батюк А.Є., Цмоць І.Г. Методи синтезу спеціалізованих обчислювальних систем для розв’язання задач у реальному часі / Інформаційні технології і системи. Т2, №1, Львів 1999 с.155-161.

51. Данілов П.О., Цмоць І.Г., Ігнатєв І.В. Апаратна реалізація обчислення максимального і мінімального чисед в масиві даних/ Сучасні комп’ютерні інформаційні технології.