УДК 621.398:007

В.П. КОЖЕМ'ЯКО, А.А. ЯРОВИЙ, Ю. С. БОГОМОЛОВ, К. Ю. ВОЗНЕСЕНСЬКИЙ

МЕТОДОЛОГІЧНІ ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ НЕЙРОПОДІБНИХ ПАРАЛЕЛЬНО-ІЄРАРХІЧНИХ СИСТЕМ НА ОСНОВІ ТЕХНОЛОГІЙ GPGPU

Вінницький національний технічний університет,
Хмельницьке шосе, 95, Вінниця, 21021, Україна,
тел.: +380 (432) 580019, E-mail: axa@vinnitsa.com

Анотація. В роботі розглянуто методологічні особливості реалізації нейроподібних паралельно-ієрархічних систем на основі технологій GPGPU, в контексті дослідження програмно-апаратної платформи GPGPU. В дослідженнях здійснено аналіз архітектури сучасних GPU та моделі паралельного програмування GPU, а також розглянуті способи оптимізації GPU програм для паралельних обчислень загального призначення в паралельно-ієрархічних системах, які запропоновані у вигляді мережевої моделі нейроподібної схеми обробки інформації.

Аннотация. В работе рассмотрены методологические особенности реализации нейроподобных параллельно-иерархических систем на базе технологий GPGPU, в контексте исследования программно-аппаратной платформы GPGPU. В исследованиях произведен анализ архитектуры современных GPU и модели параллельного программирования GPU, а также рассмотрены способы оптимизации GPU программ для параллельных вычислений общего назначения в параллельно-иерархических системах, которые предложены в виде сетевой модели нейроподобной схемы обработки информации.

Abstract. The methodological features of realization of neural-like parallel-hierarchical systems on the basis of technologies GPGPU, in a context of research of a hardware-software platform of GPGPU are considered in the work. The analysis of architecture of modern GPU and model of parallel programming GPU is carry out in researches. Also considered the methods of the GPU programs optimization for parallel computing of general purpose in parallel-hierarchical systems which are offered as network model of the neural-like scheme of information processing.

Ключові слова: паралельно-ієрархічні нейроподібні системи, паралельні обчислення, GPGPU, програмування відеоадаптерів, обробка зображень.

ВСТУП

Стрімкий перехід сучасних систем управління на цифрові стандарти привів до необхідності обробляти з високою швидкістю надвеликі обсяги інформації. Складна обробка й фільтрація сигналів, наприклад розпакування стислих аудіо- та відеоданих, маршрутизація інформаційних потоків, класифікація і прогнозування швидкозмінних динамічних даних у реальному часі вимагає застосування досить продуктивних інтелектуальних обчислювальних систем. Подібні системи можуть бути реалізовані в різноманітних методологічних базисах та на різноманітній елементній базі, але найбільшого поширення на сучасному етапі одержали паралельні технології, зокрема паралельні нейроподібні мережеві пристрої [1-3].

В роботі досліджуються та аналізуються технології паралельних обчислень в контексті їх застосування для різноманітних задач розпізнавання образів та обробки зображень, які також потребують ефективно оперувати неформалізованими даними, в тому числі спеціалізовані моделі паралельної обробки в образному комп’ютері. Характерною особливістю структур образного комп'ютера є те, що вони повинні оперувати не формалізованими даними, а образною інформацією, що потребує проведення паралельної обробки інформації на різних рівнях ієрархії в динаміці та реальному часі. Тому виникає ряд

© В.П. КОЖЕМ'ЯКО, А.А. ЯРОВИЙ, Ю. С. БОГОМОЛОВ, К. Ю. ВОЗНЕСЕНСЬКИЙ, 2008

задач зі створення таких мережевих обчислювальних структур, які б здійснювали паралельну обробку даних (образів) та проводили їх аналіз (деталізацію) на різних рівнях ієрархії. Такі структури отримали назву паралельно-ієрархічних і пропонуються як перспективний формальний апарат для побудови спеціалізованих моделей образного комп'ютера [4,5]. Паралельно-ієрархічні системи застосовуються для виділення характерних ознак зображень, їх кодування і скорочення розмірності при виконанні паралельних обчислень. Гарна збіжність паралельно-ієрархічного перетворення використовується в структурах паралельної пам'яті, аналізі і розпізнаванні зображень, при кодуванні і стисненні даних, а також для обробки біомедичних сигналів [5].

Прикладом паралельно-ієрархічних пірамідальних структур може бути названий клас мульти-SIMD-систем, у яких декілька рівнів ідентичних процесорних елементів (ПЕ) працюють автономно в SIMD-режимі. Кожному ПЕ відповідає елемент зображення на відповідному рівні дозволу. До цієї групи відносяться такі системи: PCLIP, PAPIA, GAM, SPHINX [6]. У іншому випадку декілька потужних ідентичних процесорних блоків об'єднуються в ієрархічну пірамідальну структуру. Кожному процесорному блоку відповідає частина оброблюваного зображення. До цього сімейства належать системи: Uhr's Array/Net і EGPA [6].

Швидкодія класичних однопроцесорних комп’ютерних систем у сукупності з існуючими послідовними алгоритмами не дозволяють проводити обробку великих масивів даних у реальному часі. У зв'язку з цим останнім часом побудовані паралельні алгоритми ортогональних перетворень [7], що допускають ефективне розпаралелювання. Тут лінійні перетворення реалізуються на паралельних системах типу SIMD і MISD із довільним, заздалегідь заданим числом гілок. Для таких паралельних ортогональних перетворень розроблені методи синтезу ортогональних базисних систем, перетворення у яких мають більш швидкий алгоритм обчислення.

МЕТА ДОСЛІДЖЕННЯ

Метою роботи є дослідження методологічних особливостей реалізації нейроподібних паралельно-ієрархічних систем на основі технологій GPGPU, в контексті дослідження програмно-апаратної платформи GPGPU, аналізу архітектури сучасних GPU та моделі паралельного програмування GPU, а також способів оптимізації GPU програм.

ПОСТАНОВКА ПРОБЛЕМИ. ОСОБЛИВОСТІ СТРУКТУРНО-ФУНКЦІОНАЛЬНОЇ ОРГАНІЗАЦІЇ НЕЙРОПОДІБНИХ ПАРАЛЕЛЬНО-ІЄРАРХІЧНИХ СИСТЕМ

Мережеве перетворення є одним із шляхів реалізації паралелізму і дозволяє формалізувати процедури паралельної взаємодії багаторівневої організації обчислень у часі на різних рівнях ієрархії.

Принцип побудови пірамідальної ієрархічної структури даних, можна визначити як послідовність масивів даних того самого інформаційного поля на різноманітних рівнях дозволу: $P=(A_0, A_1, A_2, \dots, A_L)$, де A_i – інформаційне поле, i – номер рівня дозволу, $i=\overline{0,l}$. Така піраміда інформаційних полів формує обчислювальну структуру, яка дозволяє у реальному часі реалізувати методи інтелектуального сенсорного сприйняття. Зокрема, подібна структура дозволяє керувати рівнем дозволу оброблюваних даних, а також розмірами ділянки аналізу. Розміри аналізованого "вікна" даних можуть бути постійними, але, переміщуючись з одного рівня розміщення на інший, можна робити обробку того самого елемента інформаційного поля з різним ступенем деталізації. При цьому рішення про необхідність подальшої обробки можна прийняти на верхньому рівні обробки після аналізу інформаційного поля з малим дозволом, кожний елемент якого містить інтегральні оцінки про відповідні фрагменти вихідного поля на найнижчому рівні, що призводить до підвищення швидкості їх обробки. Сутність пірамідального підходу полягає в одночасному використанні при аналізі послідовності масивів даних на різноманітних рівнях ієрархії. Це дозволяє реалізувати стратегію від "загального до часткового", що дає можливість реалізувати концепцію нейроподібної обробки. Кожний елемент піраміди інформаційного поля характеризується трьома координатами (i,j,k), де i – рядок, j – стовпчик, k – рівень [2,4].

Таким чином, принцип побудови паралельно-ієрархічної структури даних можна визначити як послідовність операцій над множинами масивів даних, що утворюють множини інформаційних полів різноманітних рівнів ієрархії, взаємодія між якими здійснюється пірамідальною ієрархічною структурою і реалізується на основі мережевої архітектури.

Структурна схема взаємодії інформаційних потоків у паралельно-ієрархічній обчислювальній структурі подана на рис. 1, в якій обробляється множина вхідних потоків даних на різноманітних (k) ієрархічних рівнях. Кожний рівень являє собою сукупність процесорних елементів (ПЕ), які функціонують

у чітко фіксовані моменти часу (t_j) [4].

Концепція формування паралельно-ієрархічної мережевої структури припускає багатоетапний процес послідовного перетворення корельованих і утворення декорельованних у часі елементів фізичного середовища при його переході з одного стійкого стану в інший.

Нехай фізичне середовище описується цифровою інформацією, яка задана у вигляді множин:

$$M_{1,1}^{0},\ldots,M_{1,k}^{0},\ldots,M_{1,N_1^0}^{0},\ \text{де}\ M_{1,k}^{0}=\left\{a_{i_1}^{k_1}\right\},$$

$$i_1=\overline{1;n_{k_1}^{0}};\qquad k_1=\overline{1;N_1^{0}};\quad a_{i_1}^{k_1}\in R;\qquad a_{i_1}^{k_1}\neq 0.$$

Ця інформація перетворюється і формується наступна модель паралельно-ієрархічної мережевої структури. З кожної множини $M_{1,k}^{0}$ в момент часу t_1 вибирається за F^* критерієм довільний елемент, позначимо його $\bigcup_{i_1=1}^{n_{k_1}^{1}}\left(a_{i_1}^{k_1}\right)^{t_1}$, кратність цього елемента позначимо $\left(r_{k_1}\right)^{t_1}$. Причому, із множини $M_{1,k}^{0}$ вибір елемента за F^* критерієм відбувається лише в тому випадку, якщо в часі складові елементи даної множини корелюють між собою [2].

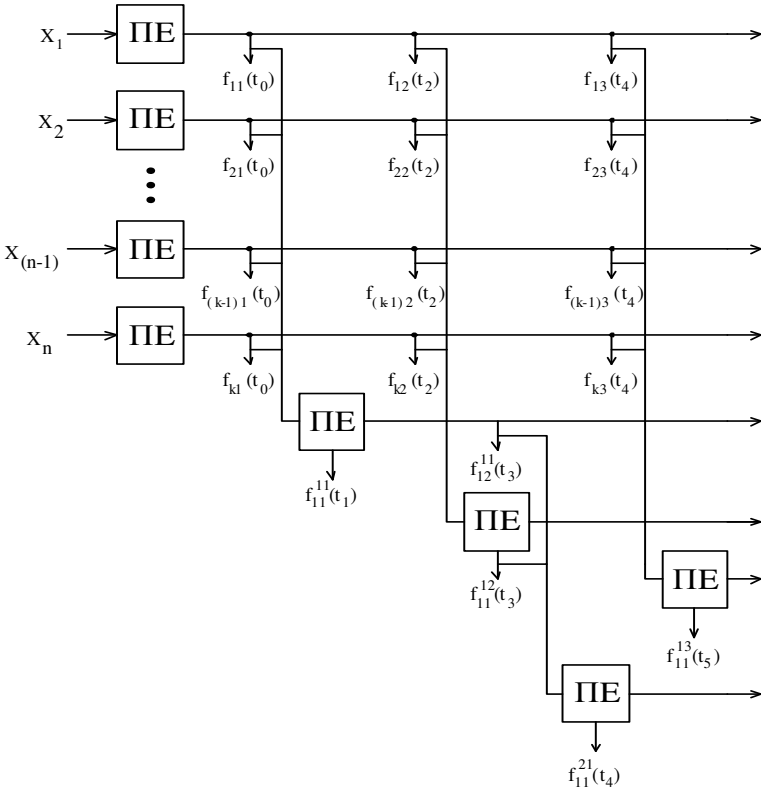


Рис. 1. Структурна схема взаємодії інформаційних потоків у паралельно-ієрархічній мережевій структурі

Надалі при описі паралельно-ієрархічної мережевої структури передбачається, що при виборі на кожному етапі довільного елемента з відповідної множини складові його елементи також корелюють у часі.

Паралельно-ієрархічна обробка дозволяє перетворити вихідну цифрову інформацію, задану у вигляді N_1^0 множин, і в чітко фіксовані моменти часу представити її як паралельно-ієрархічну мережеву структуру. Починаючи з другого рівня, запам'ятовуються ті добутки, що декорельовані в часі з іншими проміжними результатами. Останні визначаються кореляцією взаємодіючих елементів. Елементи паралельно-ієрархічної мережевої структури, що декорельовані в часі [2,4]:

$\left(r_{k_2=1}\right)^{t_2}\bigcup_{i_2=1}^{n_{k_2=1}^{1}}\left(a_{i_2}^{1}\right)^{t_2}$ – на 2-му рівні, $\left(r_{k_3=1}\right)^{t_5}\bigcup_{i_3=1}^{n_{k_3=1}^{1}}\left(a_{i_3}^{1}\right)^{t_5}$ – на 3-му рівні, ... , $\left(r_{k_{j+1}=1}\right)^{t_{3j-1}}\bigcup_{i_{j+1}=1}^{n_{k_{j+1}=1}^{1}}\left(a_{i_{j+1}}^{1}\right)^{t_{3j-1}}$ – на $(j+1)$-му рівні. В загальному вигляді математична модель паралельно-ієрархічної мережевої структури набуває такого вигляду:

$$\sum_{j=1}^{n}\left(r_{k_{j+1}=1}\right)^{t_{3j-1}}\bigcup_{i_{j+1}=1}^{n^1_{k_{j+1}=1}}\left(a^1_{i_{j+1}}\right)^{t_{3j-1}}=\sum_{k_1=1}^{N^0_1}\sum_{i_1=1}^{n^0_{k_1}}a^{k_1}_{i_1},$$

де k - кількість рівнів паралельно-ієрархічного перетворення [2,4].

АНАЛІЗ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ НА GPU ТА ЇХ ОПТИМІЗАЦІЯ

Наукові дослідження концентруються на розробці нейроемулятора – системи, побудованої на базі каскадного з’єднання універсальних SISD-, SIMD- або MISD-процесорів (наприклад, Intel, AMD, Sparc, Alpha, Power PC та ін.), яка реалізує типові нейрооперації (зважене підсумовування й нелінійне перетворення) на програмному рівні. Вирішення поставленої задачі можливе лише за умови коректного та обґрунтованого вибору нейроприскорювача – нейрокомп’ютера, реалізованого у вигляді карти або модуля з розпаралелюванням операцій на апаратному рівні або ж конструктивно-автономної системи. Нейрокомп’ютери, виготовлені у вигляді карт (віртуальні нейрокомп’ютери), як правило, призначені для встановлення у слот розширення комп’ютерної системи (стандартного ПК).

На основі проведених наукових досліджень [8,9], пропонується в якості нейроприскорювача, як апаратної платформи для реалізації масштабних нейронних та нейропобідних паралельно-ієрархічних мереж, обрати технологію GPGPU – General-Purpose computation on Graphic Processing Units (обчислення загального призначення на відеоадаптерах). GPGPU – це по суті використання потужного відеоадаптера для виконання спеціалізованих, в тому числі паралельних, обчислень, які зазвичай виконуються на CPU в ПК. Оскільки сучасні технології побудови відеоадаптерів дозволяють використання 128-ядерних спецпроцесорів, в порівнянні із сучасними 4-ядерними мультимедійними CPU, то застосування їх для нейроемуляції різних топологій масштабних нейронних та нейропобідних паралельно-ієрархічних мереж є актуальним та перспективним.

В контексті програмної реалізації здійснюється робота із створення нейропакета для реалізації різних топологій масштабних нейронних та нейропобідних паралельно-ієрархічних мереж і можливості прорахунку їх на GPU (реалізація процесів паралельно-ієрархічної обробки інформації та відповідних методів навчання). Під топологією розуміємо певний набір шарів мережі, відповідно пов'язаних між собою. Кількість шарів мережі, зв'язки між ними, кількість нейронів у шарі, функція активації шару, зв'язки між нейронами різних шарів задаються користувачем. Одна із ключових можливостей програмного продукту – гнучкість у створенні топології; що робить можливим реалізацію масштабних паралельно-ієрархічних та ієрарх-ієрархічних нейропобідних мереж [8].

Таким чином, в даній роботі не ставилась задача детально описати організацію відеоадаптерів на схемотехнічному рівні. В роботі досліджуються методологічні особливості архітектури відеоадаптерів для програмної реалізації паралельних обчислень (обчислень загального призначення) на низькому рівні програмування.

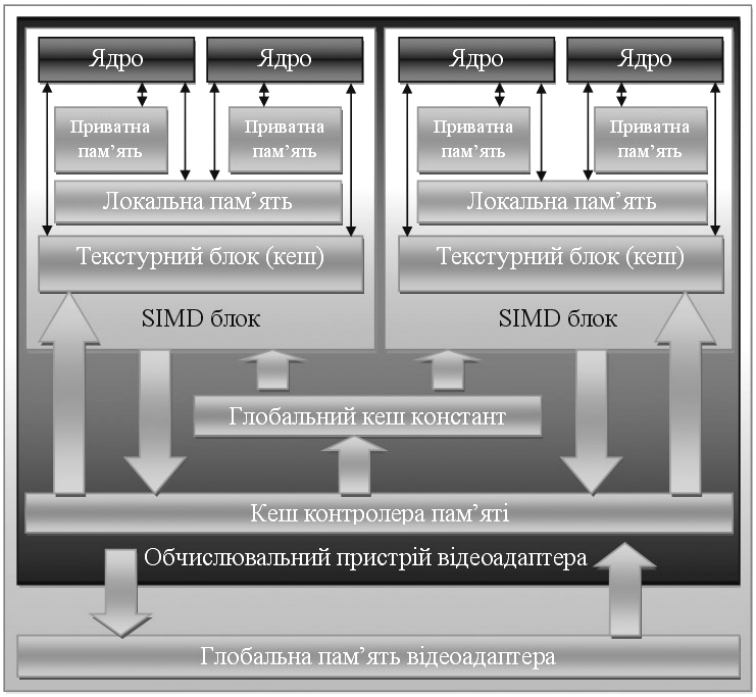


Рис. 2. Узагальнена модель відеоадаптера

На рис. 2 зображена спрощена модель відеоадаптера. Кількість обчислювальних блоків у сучасних відеоадаптерах набагато більше двох (10 – для відеоадаптера ATI RadeonHD 48xx та 10 – для відеоадаптера GeForce 280GTX). Кількість ядер у кожному обчислювальному блоці також набагато вища (80 ядер – для відеоадаптера ATI RadeonHD 48xx та 24 ядра – для відеоадаптера GeForce 280GTX). Локальна пам'ять використовується лише при обчисленнях загального призначення для синхронізації потоків і реалізована не в усіх відеоадаптерах (наприклад, відсутня у ATI RadeonHD до 4-ої серії). Текстурні блоки, яких на кожен SIMD-блок припадає до 4-ох, мають власний кеш (проміжний буфер із швидким доступом, що містить копію тієї інформації, яка зберігається в пам'яті з менш швидким доступом, але з найбільшою ймовірністю може бути звідти запитаний) та дозволяють завантажувати дані з глобальної пам'яті в приватну, яка належить лише конкретному ядру SIMD-блока („керуючи" ці дані у власному кеші). Усі операції зчитування та запису у глобальну пам'ять виконуються через контролер пам'яті і таким чином зберігаються у кеш контролера пам'яті [10].

Таким чином, принцип паралелізму реалізовано в сучасних відеоадаптерах на багатьох рівнях. У паралельно працюючих над різними задачами (потоками інструкцій) SIMD-блоках виконується паралельна обробка даних однієї задачі. У сучасних відеоадаптерах зазвичай використовується від 2 до 8 контролерів пам'яті, що дозволяє збільшити пропускну спроможність пам'яті.

Узагальнена модель програмування GPU.

Перед виконанням програми її константи завантажуються у „глобальний кеш констант". Після цього спеціальний диспетчер виділяє певній програмі певну кількість SIMD-блоків, на яких вона буде виконуватися. Усі ядра у SIMD-блоці виконують одні і ті самі операції, але над різними операндами (single instruction multiply data), тобто розділяють потік інструкцій. Кожне ядро одночасно може обробляти декілька потоків, кількість яких обмежена лише розміром локальної пам'яті ядра. Під одночасною обробкою розуміється одночасне розміщення даних у пам'яті та почергове виконання інструкцій з цих потоків, що дозволяє маскувати затримки при зчитуванні з глобальної пам'яті.

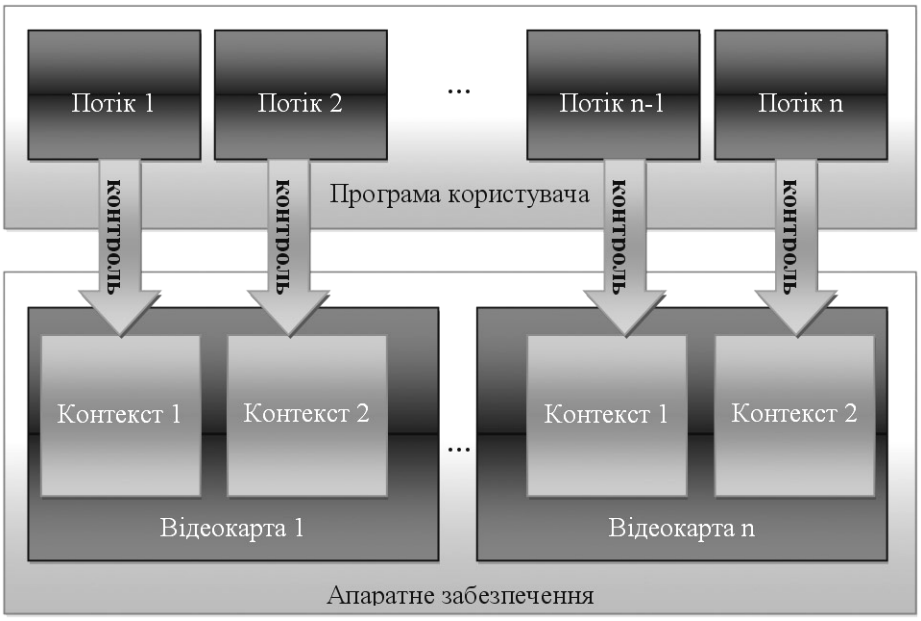


Рис. 3. Узагальнена модель програмування GPU

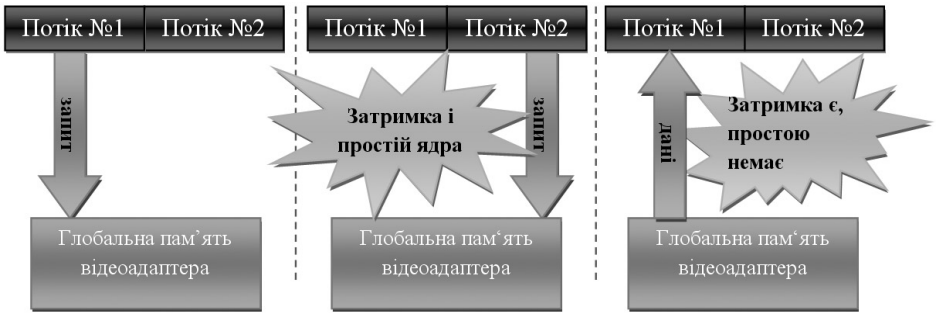
На рис. 3 зображено узагальнену модель програмування відеоадаптерів. Усі сучасні платформи API (Application Programming Interface) для програмування відеоадаптерів підтримують можливість використання декількох відеоадаптерів у межах однієї системи (SLI у NVidia, Crossfire/CrossfireX у AMD/ATI). Для кожного відеоадаптера можна призначити декілька контекстів виконання, і з кожним контекстом асоціюється один потік операційної системи. В рамках контексту створюються шейдери (програми для одного із ступенів графічного конвеєра, використовувані в тривимірній графіці для визначення остаточних параметрів об'єкта або зображення) та виділяється пам’ять під дані користувача (текстури у графічній інтерпретації). Одночасно на одному відеоадаптері може виконуватись лише один шейдер із одного контексту. Процеси виклику і передачі даних є асинхронними [10].

Способи оптимізації GPU програм.

1. Маскування затримок при звертанні до глобальної пам’яті.

$$T_{\text{загальна затримка}} = N_{\text{завантажень}} \times \tau_{\text{затримки}}$$

Рис. 4. Цикл передачі та обробки даних без маскування затримок



$$T_{\text{загальна затримка}} = \tau_{\text{затримки}}$$

Рис. 5. Цикл передачі та обробки даних з маскуванням затримок

На рис. 4 та рис. 5 наведено спрощену схему маскування затримок при звертанні до глобальної пам'яті. Потоки 1 та 2 виконуються на одному ядрі. За відсутності маскування затримок для кожного потоку відбувається наступна процедура:

1) виконується запит до пам'яті;

2) ядро очікує на дані протягом 32 тактів (називається затримка, або простій ядра);

3) коли надходять дані із пам'яті, виконується їх обробка, і керування передається наступному потоку.

За використання схеми маскування затримок відбуваються такі дії:

1) послідовно виконуються запити від всіх потоків;

2) ядро очікує на дані протягом 32 тактів після першого запиту (одноразова затримка);

3) дані надходять із пам'яті безперервно у порядку їх запиту, передача керування між потоками відбувається миттєво, без додаткових затримок.

Таким чином, за використання схеми маскування затримок максимальний час простою ядра рівний 32 тактам [10].

*2. Маскування затримок при копіюванні даних з/на GPU.*Для відеоадаптерів Nvidia GeForce 2xx та ATI RadeonHD 2xxx можливі асинхронні операції, тобто можливо одночасно виконувати обчислення та копіювати дані з/на відеоадаптер. Це дозволяє мінімізувати затримки при виконанні програм, що викликають виконання декількох шейдерів на відеоадаптерах. Тобто можливо одночасно виконувати один шейдер та завантажувати дані, необхідні для наступного [10].

*3. Послідовне зчитування із пам'яті.*Як показують результати тестів [11], найкращим за часом звертання способом завантаження даних з глобальної пам'яті є послідовне зчитування даних. Одним з кращих способів організації послідовного зчитування даних є зчитування даних за адресою, яка визначається функцією від номеру потоку [10].

*4. Використання локальної пам'яті та текстурного кешу.*При створенні алгоритму обробки даних, який потрібно виконати на відеоадаптері, слід намагатися поділити дані на невеликі порції, які могли б бути розміщені у локальній пам'яті і використовуватися усіма потоками, що обробляються на SIMD-блоці, та розміщувати потоки, що суміжно використовують певні дані у групах потоків для AMD StreamSDK або блоках NVidia CUDA, оскільки дані, що зчитуються сусіднім потоком, зазвичай залишаються у текстурному кеші, через який найчастіше виконується завантаження даних [10].

*5. Узгодженість потоків в рамках одного SIMD-блоку.*Для маскування затримок необхідно запускати на SIMD-блоці певну кількість потоків, при чому мінімум у 4 рази більше, ніж кількість ядер у SIMD-блоці. Кожне розгалуження алгоритму шейдера призводить до поділу потоку інструкцій на два менших потоки відповідно до гілок. Після цього ці дві групи виконуються послідовно. Такий поділ потоку інструкцій є небажаним явищем, оскільки він призводить до неповної утилізації ресурсів відеоадаптера (не досягає максимальної ефективності), проте уникнути його можливо лише у шейдері без будь-яких розгалужень. Можливі шляхи мінімізації впливу явища поділу потоку інструкцій – це особлива організація вхідних даних залежно від групи потоків, або написання розгалужень таким чином, щоб умова розгалуження залежала від номеру групи потоків. Також можна запускати набагато більшу кількість потоків на SIMD-блок (ця кількість повинна обчислюватись для відеоадаптера виходячи з кількості SIMD-блоків) [10].

ВИСНОВКИ

Таким чином, в роботі розглянуто методологічні особливості реалізації нейроподібних

паралельно-ієрархічних систем на основі технологій GPGPU, зокрема, в контексті дослідження програмно-апаратної платформи GPU. Здійснено аналіз архітектури сучасних GPU та моделі паралельного програмування обчислень загального призначення на відеоадаптерах, а також розглянуті способи оптимізації GPU програм для паралельних обчислень в паралельно-ієрархічних системах, які запропоновані у вигляді мережевої моделі нейроподібної схеми обробки інформації. Результати проведених досліджень показують перспективу застосування технологій GPGPU та відеоадаптерів як графічних нейроприскорювачів для підвищення ефективності обробки надвеликих обсягів даних в задачах розпізнавання образів та обробки зображень, особливо із застосуванням нових підходів до програмування сучасних поколів відеоадаптерів, що дозволяє абстрагуватися від графічно-орієнтованої природи функціонування відеоадаптерів до використання GPU як системи SIMD-процесорів з високим ступенем паралелізму.

В подальших дослідженнях планується удосконалити структурно-функціональну організацію паралельно-ієрархічних інтелектуальних систем на основі технологій GPGPU, для реалізації різноманітних рівнів розпаралелювання в структурах образного комп'ютера.

Особливо ж перспективною представляється ідея реалізації паралельно-ієрархічних структур на основі оптико-електронних елементів і пристроїв із динамічною багатофункціональністю [2,4]. Саме завдяки сполученню ідей паралельно-ієрархічної алгоритміки [2] і оптико-електронної нетрадиційної схемотехніки [12] є можливим створення ефективних (у тому числі нейроподібних) обчислювальних і інформаційних структур, у яких розширення функціональних і інтелектуальних можливостей технічних і алгоритмічних засобів на різноманітних рівнях ієрархії випереджає ріст їхньої складності.

СПИСОК ЛІТЕРАТУРИ

В.В.Воеводин Параллельные вычисления : учебн. пособие [для студ. высш. учебн. зав.] / В.В.Воеводин, Вл.В.Воеводин – СПб.: БХВ-Петербург, 2002. – 608 с. – ISBN 5-94157-160-7.

1. Паралельно-ієрархічне перетворення як системна модель оптико-електронних засобів штучного інтелекту : [Монографія.] / В.П. Кожем’яко, Ю.Ф. Кутаєв, С.В. Свєчніков, Л.І. Тимченко, А.А. Яровий – Вінниця: УНІВЕРСУМ-Вінниця, 2003. – 324 с. – ISBN 966-641-072-9.
2. Апаратна реалізація паралельно-ієрархічної мережі на основі DSP / Кожем'яко В.П., Тимченко Л.І., Яровий А.А., Ремезюк С. : збірник тез доповідей третьої міжнародної науково-технічної конференції [Оптоелектронні інформаційні технології “Фотоніка ОДС–2005”], (Вінниця, 27-28 квітня 2005 р.) – Вінниця: „УНІВЕРСУМ-Вінниця”, 2005. – С. 43.
3. В.П. Кожем’яко Паралельно-ієрархічні мережі як структурно-функціональний базис для побудови спеціалізованих моделей образного комп’ютера : [Монографія.] / В.П. Кожем’яко, Л.І. Тимченко, А.А. Яровий. – Вінниця: Універсум-Вінниця, 2005. – 161 с. – ISBN 966-641-142-3.
4. Образний відео-комп’ютер око-процесорного типу : [Монографія.] / Кожем’яко В.П., Лисенко Г.Л., Яровий А.А., Кожем’яко А.В. – Вінниця: Універсум-Вінниця, 2008. – 215 с. – ISBN 978-966-641-261-7.
5. Иерархическая обработка изображений и пирамидальные системы / Воробьев К.Ю., Тимонькин Г.Н., Харченко В.С., Мельников В.А. // Зарубежная радиоэлектроника. - 1991. - №7. - С.51- 61.
6. СБИС для распознавания образов и обработки изображений : учебн. пособие [для студ. высш. учебн. зав.] / Под ред. К.Фу. – М.: Мир, 1988. – 248 с.
7. Вибір апаратної платформи для реалізації масштабних нейронних та нейроподібних паралельно-ієрархічних мереж [Електронний ресурс] : (IX Міжнародна конференція Контроль і управління в складних системах (КУСС-2008), Вінниця, 21-24 жовтня 2008 року) / А.А. Яровий, Ю.С. Богомолов, К.Ю. Вознесенський – Режим доступу: http://www.vstu.vinnica.ua/mccs2008/materials/subsection_2.2.pdf.
8. Сравнение производительности графических ускорителей и центрального процессора при вычислениях для больших объемов обрабатываемых данных / Скрибцов П.В., Долгополов А.В. // Нейрокомпьютеры: разработка, применение – Москва, Издательство "Радиотехника", 2007. – № 9. – С. 421-425. – ISSN 0869-5350.
9. GPGPU: General Purpose computations on Graphic Processing Unit [Електронний ресурс] – Режим доступу: http://www.gpgpu.org.
10. Набір мікротестів швидкодії відеокарт у задачах загального призначення [Електронний ресурс] – Режим доступу: http://graphics.stanford.edu:80/projects/gpubench.
11. Кожемяко В.П. Оптоэлектронные логико-временные информационно-вычислительные среды /

Кожемяко В.П. – Тбилиси, Мецниэреба, 1984. – 357 с. – Библиогр.: 146, с. 343-355.

Надійшла до редакції 26.09.2008 р.

КОЖЕМ'ЯКО В.П. – академік АІНУ, д.т.н., професор, завідуючий кафедрою лазерної і оптоелектронної техніки, Вінницький національний технічний університет, Вінниця, Україна.

ЯРОВИЙ А.А. – к.т.н., доцент кафедри інтелектуальних систем, науковий співробітник кафедри лазерної і оптоелектронної техніки, Вінницький національний технічний університет, Вінниця, Україна.

БОГОМОЛОВ Ю.С. – студент кафедри інтелектуальних систем, Вінницький національний технічний університет, Вінниця, Україна.

ВОЗНЕСЕНСЬКИЙ К.Ю. – студент кафедри інтелектуальних систем, Вінницький національний технічний університет, Вінниця, Україна.