prof. Musiyenko M.P., Krainyk Ya.M.

MINIMAL SUM ALGORITHM PARALLEL IMPLEMENTATION FOR LDPC-DECODER

The aim of the article was rising up throughput of LDPC-decoder. It is based on parallel computation organization according to minimal sum algorithm.

This work investigates minimal sum algorithm that is one of the soft decoding algorithms. Parallel model of decoding stages implementation is provided for partial parallel decoder case. Parameters that have influence on throughput are researched. Computation organization model is designed. The model is implemented with use of FPGA integrated circuit and comparison between parallel and sequential model is performed.

Keywords: LDPC-codes, minimal sum algorithm, LDPC-decoder, parallel execution.

УДК 004.315.7

д.т.н., проф. **Мясіщев О.А.** (ХмНУ)
Варчак Д.Ю. (ХмНУ)
Лєнков О.С. (ВІКНУ)

ВИКОРИСТАННЯ ТЕХНОЛОГІЙ GPGPU ПРИ РОЗВ'ЯЗАННІ СИСТЕМИ ЛІНІЙНИХ РІВНЯНЬ

По причині недостатньої швидкодії CPU в задачах, які в кінцевому результаті приводяться до рішення системи лінійних рівнянь, виконаний аналіз технологій GPGPU таких, як CUDA та Stream Technology. Виділено основні параметри за допомогою яких можна оцінити графічні адаптери, і здійснено їх порівняльну характеристику.

Ключові слова: Stream Technology, CUDA, BLAS, GPGPU

Вступ. Сьогодні новини про використання графічних процесорів для загальних обрахунків можна почути на кожному кроці. Ріст швидкодії процесорів сильно загальмувався, тому з'явилися такі вирази, як CUDA, Stream Technology та OpenCL і стали одними з найбільш цитованих в сфері IT. Однак значення цих виразів і технологій, які за ними стоять, мало відомі.

Велика кількість задач зводяться до рішення систем лінійних рівнянь та матричних розрахунків. Сюди відносяться розрахунки теплопровідності, пружності, міцності та інші інженерні задачі, які вимагають великої продуктивності від пристрою на яких вони проводяться. Останніми роками щоб максимально збільшити швидкість обрахунку тої чи

іншої задачі, застосовують паралелізм. На даний момент вирішення для поставлених задач використовують багатоядерні процесори та графічні адаптери. Саме тут і використовуються розробленні технології GPGPU (General-purpose graphics processing units, Графічний процесор загального призначення), а саме CUDA, Stream Technology та OpenCL.

GPGPU– техніка використання графічного процесору для розрахунків загального призначення [1]. Це стало можливим завдяки додаванню програмованих шейдерних блоків що дозволяє розробникам програмного забезпечення використовувати потокові процесори для неграфічних даних.

На ринку інформаційних технологій представлена велика кількість графічних адаптерів, які надають можливість вирішувати описані вище задачі, зокрема корпорацій AMD та Nvidia.

Постановка задачі. Для ефективного та швидкого вирішення ряду інженерних завдань доцільно використовувати графічні адаптери. Отже метою статті є аналіз існуючих технологій GPGPU, а саме CUDA та Stream Technology. Порівняння параметрів та швидкодії по обраних критеріях графічних адаптерів, а також співставлення їх з аналогічними параметрами багатоядерного процесора. В статті розглянуто основні принципи за якими виконуються обрахунки на графічних картах.

Огляд технології CUDA. Технологія CUDA – технологія GPGPU, яка дозволяє програмістам реалізовувати мовою програмування С алгоритми, що виконуватимуться на графічних процесорах GeForce восьмого покоління і вище, для вирішення складних обчислювальних завдань за менший час завдяки багатоядерній обчислювальній потужності графічних процесорів [2].

Технологія CUDA передбачає, що програміст спочатку розбиває задачу на незалежні частини – блоки, які можуть виконуватись паралельно. Опісля кожен блок розбивається на багато потоків, які паралельно виконуються, і які можуть залежати один від одного. CUDA забезпечує засоби розширення мови С для паралельного запуску великої кількості потоків, які виконують одну, і ту ж функцію. Потоки об'єднуються в блоки, до 512 в кожному, а блоки, в свою чергу – в сітки. Потоки всередині блоку запускаються на одному мультипроцесорі, мають спільну поділювану пам'ять і можуть синхронізувати хід виконання задачі. Кожен потік має унікальний ідентифікатор всередині блоку, що виражається за допомогою одномірного, двомірного або тьохмірного індексу. Розмірність блоку доступна через вбудовану зміну blockDim. Максимальні розмірності: 512, 512, 64. Решітки можуть бути одномірними або двомірними, максимальне значення індексу: 65535. Індекс блоку в решітці доступний через вбудовану зміну blockIdx. Компоненти індексів нумеруються з нуля.

Порядок виконання блоків не визначений, блоки повинні бути незалежні один від одного. При запуску ядра, блоки решітки нумеруються і розподіляються по MP (MultiProcessor), які мають достатньо вільну ємність регістрів поділюваної пам'яті і ресурсів планувальника команд. MP складається з 8 простих процесорів, по 2 процесора: для складних операцій (таких як множення), пула регістрів, поділюваної пам'яті та планувальника команд. Планувальник команд послідовно розбиває потоки активного блоку на порції (warp), по 4 на кожний простий процесор і виконує по одній простій команді одночасно для всіх потоків порцій за 4 цикли. Для виконання однієї команди порції потоків MP повинен завантажити операнди для всіх потоків порцій, виконати команду, записати результат. Якщо доступ до порції викликає затримку, то планувальник може перейти до наступної порції. Всі потоки порцій починають виконання програми з одної і тої ж адреси, але кожен простій процесора має власний лічильник команд і регістр стану, що дозволяє виконати умовне виконання і розгалуження. Однак, кожна гілка умови виконується всіма потоками порцій по черзі. Ті потоки, для яких умови гілки не виконуються, «пропускають хід». По закінченню розходження, всі потоки порцій знову одночасно виконують корисну роботу. Таким чином, розгалуження при виконанні всередині порції сильно уповільнює

роботу ядра. По закінченню всіх потоків блоку, ресурси МР звільняються і на нього можу бути розподілений наступний блок.

Планувальник МР має обмеження:

−по максимальному числу одночасно запущених блоків (до 8);

−максимальному числу порцій (до 24 або 32 активних порцій в різних версіях обладнання);

−максимальному числу потоків (до 768 або 1024 активних потоків в різних версіях обладнання).

Кількість потоків в блоці і кількість блоків в решітці вибирається програмістом виходячи з потреби максимально задіяти ресурси МР і з урахуванням апаратних обмежень (кількості регістрів, поділюваної пам'яті). Кількість потоків в блоці повинна бути кратна розміру порції (32).

Огляд Stream Technology. ATI Stream Technology використовує обчислювальну потужність графічних процесорів для забезпечення високої продуктивності паралельних обчислень [3]. Нижче наведено огляд моделі програмування ATI Stream Technology.

ATI Stream Technology включає в себе програмний стек і процесори ATI Stream. На рис. 1 проілюстровано взаємозв'язок компонентів ATI Stream Technology.

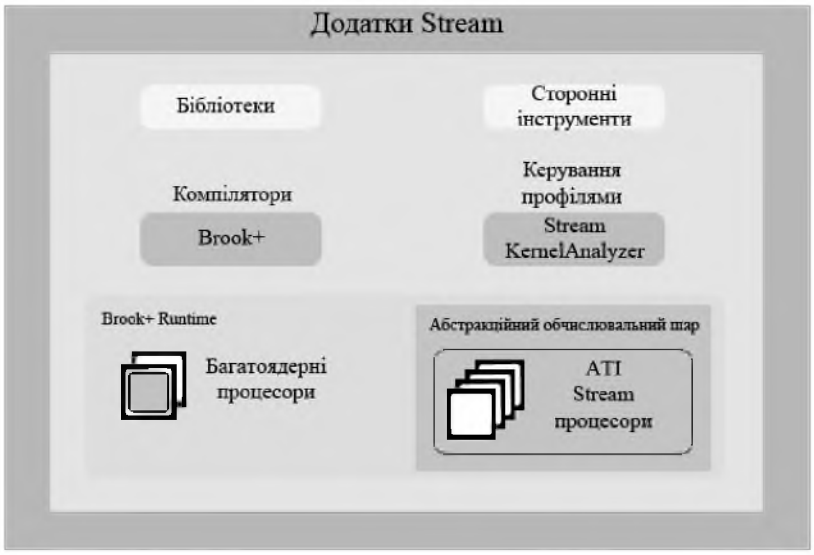

Рис. 1. Структура програмного забезпечення ATI Stream

Стек програмного забезпечення ATI Stream Technology надає кінцевим користувачам і розробникам, повний, гнучкий набір інструментів для математичних обрахунків на процесорах ATI Stream. Програмне забезпечення ATI використовує стандарти відкритих систем та відкритих платформ. Стратегія відкритої платформи ATI дозволяє партнерам розробляти та поширювати сторонні інструменти для розробки.

Програмне забезпечення включає в себе наступні компоненти:

• Компілятори - такі як компілятор Brook + з розширеннями для пристроїв ATI.

• Драйвер для потокових процесорів - ATI Compute Abstraction Layer (CAL).

• Засоби профілювання продуктивності - Stream KernelAnalyzer.

• Продуктивні бібліотеки - AMD Core Math Library (ACML) для оптимізованих предметно-орієнтованих алгоритмів.

Останнє покоління процесорів ATI Stream програмуються з використанням єдиної шейдерної моделі [4]. Програмовані потокові ядра, які задіяні у програмах, що розроблені користувачами - це так звані stream kernels. Ці потокові ядра можуть виконувати функції відмінні від графічних, за допомогою віртуальної моделі програмування SIMD, яка працює на потоках даних. У цій моделі програмування, відомої як stream computing, масиви вхідних елементів даних, які зберігаються в пам'яті, відображаються у вигляді ряду SIMD машин, які

використовують kernel для створення одного або більше виходів, які описані як вхідні масиви в пам'яті.

Кожен екземпляр kernel працює на потоковому процесорі SIMD машини і називається потоком. Зазначена прямокутна область вихідного буфера, до якого направлені потоки відома, як область виконання.

Процесор stream розпланувує масив потоків в групу потокових процесорів, поки всі потоки не будуть оброблені. Подальші kernels можуть бути виконані, поки програма не закінчить свою роботу. Спрощений вигляд моделі програмування ATI Stream Technology і відображення потоків направлених на потокові процесори показано на рис. 2.

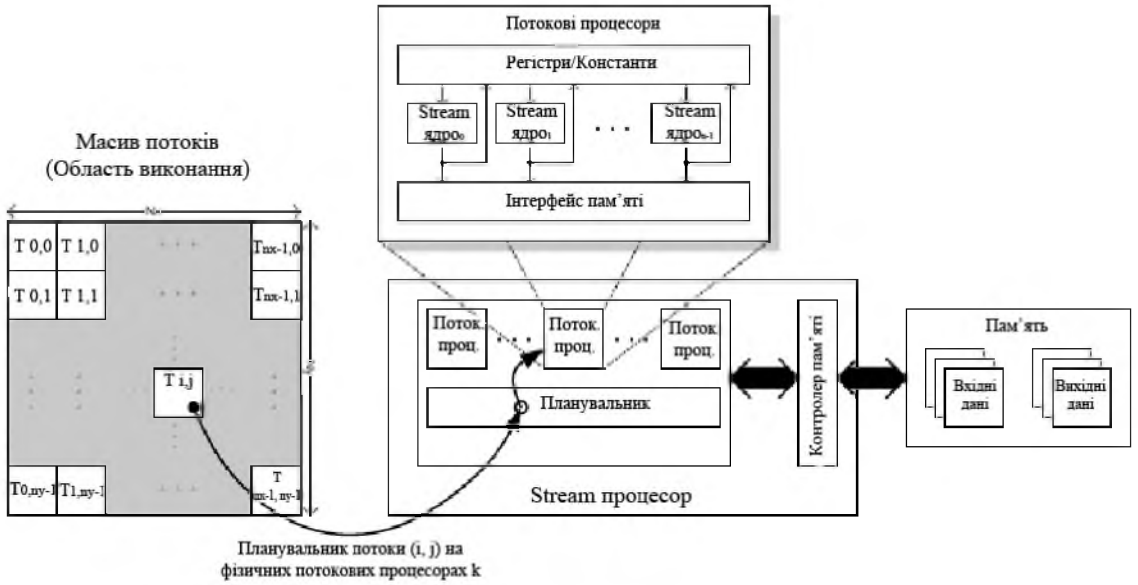

Рис. 2. Спрощена Stream Technology модель програмування

Для здійснення порівняння даних технологій, обрано 2 графічних адаптера і один процесор, які на сьогоднішній день за своїми характеристиками здатні забезпечити високу швидкодію у всіх поставлених завданнях. Від Nvidia обрано графічний адаптер GeForce 680 GTX, від AMD - Radeon HD7970. Процесор обрано Core i7 3770 від Intel. Дата релізу пристроїв відбулася в межах одного періоду часу [5, 6, 7]. В Табл.1 представлені характеристики пристроїв, від яких залежить швидкодія графічних адаптерів.

Таблиця 1

Характеристики пристроїв, які впливають на продуктивність обрахунку задач

Модель\Х-ки	Процесор	Тех. процес виконання, нм.	К-сть транзисторів, млн.	Кількість ядер	Розрядність шини пам'яті, біт.	Потужність в режимі навантаженні, ват.	Ціна на момент релізу, дол.. США.
Radeon HD7970	Tahiti XT	28	4312	2046	384	250	550
GeForce 680 GTX	GK104	28	3540	1536	256	195	500
Intel Core i7 3770	Ivy Bridge	22	1400	4	64	77	305

Для порівняння швидкодії тестування проводилося в підпрограмах SGEMM та DGEMM з бібліотеки BLAS (Basic Linear Algebra Subprograms - Основні підпрограми з лінійної алгебри). Підпрограми розроблені для множення математичних матриць зі звичайною точністю (SGEMM) та подвійною (DGEMM). Для отримання максимальних швидкодій взято квадратні матриці кратні 1024, а саме 4096. Результати були порівняні зі значеннями пікових швидкодій, які вказані в таблиці 2.

Таблиця 2

Пікові та реальні швидкодії звичайного та графічних процесорів

	Пікова швидкодія (одинарна точність), TFlops	Пікова швидкодія (подвійна точність), GFlops	Швидкодія SGEMM, GFlops	Швидкодія DGEMM, GFlops
Radeon HD7970	3,8	947	1465,31	595.67
GeForce 680 GTX	3,1	386,3	1202	126.08
Intel Core i7 3770	0,196	53,6	46,7	21,5

Проаналізувавши отримані дані, побудована гістограма (рис. 3), яка показує відношення пікових значень швидкодії та реальних, в підпрограмах SGEMM та DGEMM.

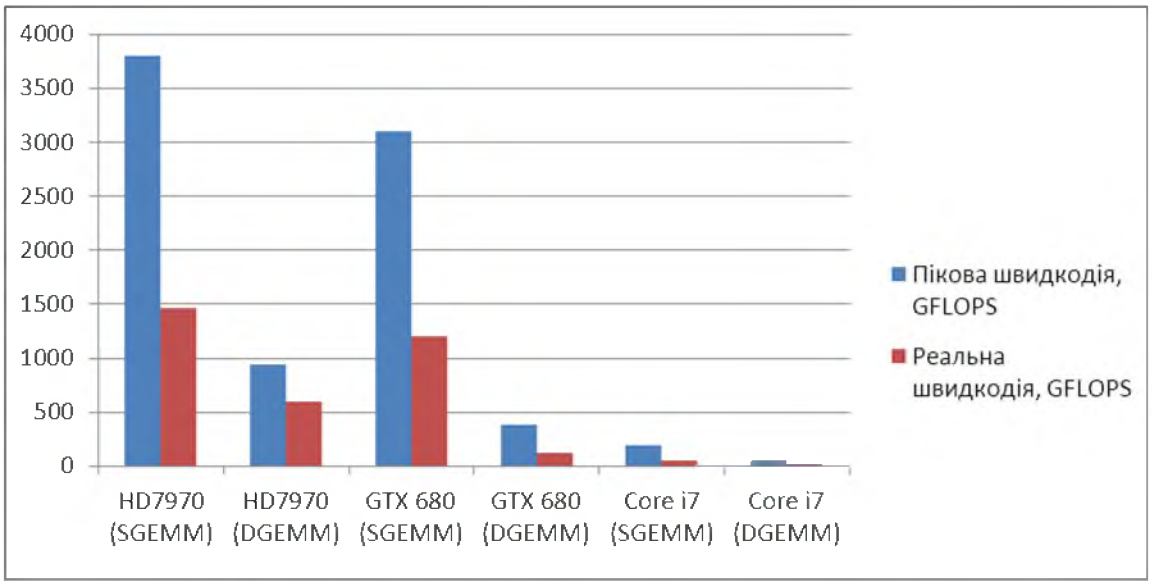

Рис. 3. Гістограма швидкодії пристроїв

Висновки. В статті висвітлено основні принципи виконання паралельних програм на графічних адаптерах при використанні технологій Stream Technology та CUDA. При вирішенні задач, які зводяться до рішення системи лінійних рівнянь доцільно використовувати GPU ніж CPU. При порівнянні швидкодії графічних адаптерів та процесора у підпрограмах для роботи з матрицями виявлено:

1. Швидкодія HD 7970 вища ніж GTX680 у 1,2 рази та 4 рази для обрахунку чисел з одинарною та подвійною точністю відповідно;

2. Швидкодія HD 7970 вища ніж Core i7 3770 у 30 разів та 26 разів, для обрахунку чисел з одинарною та подвійною точністю відповідно.

ЛІТЕРАТУРА
1. GPGPU - Вікіпедія [Електронний ресурс]. – Режим доступу : URL : http://uk.wikipedia.org/wiki/GPGPU. – Назва з екрана.
2. CUDA - Вікіпедія [Електронний ресурс]. – Режим доступу : URL : http://uk.wikipedia.org/wiki/CUDA. – Назва з екрана.
3. Technical Overview - ATI Stream Computing [Електронний ресурс]. Режим доступу : URL : http://www.student.chemia.uj.edu.pl/~mrozek/USl/wyklad/No-we_konstrukcje/Stream_Computing_Overview.pdf. – Назва з екрана.
4. John Kessenich The OpenGL® Shading Language / John Kessenich, Dave Baldwin, Randi Rost. The Khronos Group Inc., 2007.
5. Diffusion: Technology Discussion: NVIDIA GTX680 vs. ATI Radeon 7970 - SGEMM and DGEMM [Електронний ресурс]. Режим доступу : URL : http://diffusionht.blogspot.com/2013/08/nvidia-gtx680-vs-ati-radeon-7970-sgemm.html. – Назва з екрана.
6. StreamComputing | Processors that can do 20+ GFLOPS per Watt. [Електронний ресурс]. Режим доступу: URL: http://streamcomputing.eu/blog/2012-08-27/processors-that-can-do-20-gflops-watt/. – Назва з екрана.
7. CPU Performance. [Електронний ресурс]. Режим доступу : URL : http://setiathome.berkeley.edu/cpu_list.php. – Назва з екрана.

Без рецензії.

д.т.н., проф. Мясищев А.А., Варчак Д.Ю., Ленков А.С.

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ GPGPU В РАСЧЕТАХ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ

По причине недостаточной производительности CPU в задачах, которые в конечном итоге приводятся к решению системы линейных уравнений, проведен анализ технологий GPGPU таких, как CUDA и Stream Technology. Выделены основные параметры, с помощью которых можно оценить графические адаптеры, и осуществлено их сравнительную характеристику.

Ключевые слова: Stream Technology, CUDA, BLAS, GPGPU.

Prof. Myasischev A.A., Varchak D.Yu., Lenkov O.S.

USING GPGPU TECHNOLOGIES IN SOLVING THE SYSTEMS OF LINEAR EQUATIONS.

For the reason of low processing speed of CPU in multiple tasks, that in the end lead to solving the system of linear algebra, this article provides analysis of GPGPU technologies, such as CUDA and Stream Technology. Here is basic characteristics that allow us to estimate graphical adaptors and implemented their comparative characteristics in this article.

Key words: Stream Technology, CUDA, BLAS, GPGPU.