УДК 004.8

**Д.В. Гурський**
**В.О. Денисюк**

# РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ СОРТУВАННЯ BITONIC SORT

Вінницький національний технічний університет

***Анотація***

*Розглянуто реалізацію паралельного алгоритму бітонічного сортування (Bitonic Sort) за допомогою технології CUDA.Розглянуто питання аналізу існуючих методів паралельного сортування для досягнення високої продуктивності, обґрунтовано вибір засобів розробки програмного модуля та архітектури GPU, розроблено алгоритмічні схеми та блок-схеми програмного модуля, обґрунтовано вибір програмного середовища реалізації. У роботі створено програмну реалізацію паралельного алгоритму з використанням CUDA та проведено тестування його продуктивності на масивах великої розмірності. Використання результатів дозволить покращити швидкодію і продуктивність систем обробки даних, які потребують впорядкування великих об'ємів інформації.*

***Ключові слова:*** *бітонічне сортування, CUDA, паралельні обчислення, GPU, продуктивність.*

***Abstract***

*The implementation of the parallel Bitonic Sort algorithm using CUDA technology is considered. The issue of analyzing existing parallel sorting methods to achieve high performance is considered, the choice of software module development tools and GPU architecture is justified, algorithmic schemes and flowcharts of the software module are developed, the choice of the software implementation environment is justified.In the work, a software implementation of a parallel algorithm using CUDA is created and its performance is tested on large-scale arrays.Using the results will allow improving the speed and performance of data processing systems that require ordering large amounts of information.*

***Keywords:*** *Bitonic Sort, CUDA, parallel computing, GPU, performance.*

## Вступ

Актуальність реалізації паралельного алгоритму бітонічного сортування (Bitonic Sort) за допомогою технології CUDA обумовлена стрімким зростанням обсягів інформації, що потребує швидкої обробки [1-3]. Традиційні послідовні алгоритми не завжди забезпечують необхідну швидкодію для надвеликих наборів даних, що робить перехід до масово-паралельних обчислень на GPU критично важливим. Використання архітектури CUDA дозволяє значно прискорити процес сортування завдяки специфіці алгоритму Bitonic Sort, який є незалежним від вхідних даних (data-oblivious) і ідеально підходить для архітектури SIMT. Метою роботи є дослідження та програмна реалізація паралельного алгоритму бітонічного сортування для оптимізації процесу впорядкування великих масивів даних.

## Постановка задачі дослідження

Задачі дослідження полягають у вирішенні наступних питань:

- аналіз методів паралельного сортування, принципів роботи GPU та технології CUDA;
- дослідження математичної моделі бітонічного сортування та обґрунтування його переваг для розпаралелювання;
- розробка та програмна реалізація паралельного алгоритму мовою C++ із використанням CUDA Toolkit;
- тестування швидкодії алгоритму на великих наборах даних та оцінка ефективності використання GPU порівняно з послідовним виконанням.

## Виклад основного матеріалу

В основі роботи лежить алгоритм бітонічного сортування, який належить до класу сортувальних мереж. Його ключовою особливістю є незалежність послідовності порівнянь від значень елементів (data-oblivious), що робить його ідеальним для паралельної реалізації на архітектурі GPU. Теоретична складність алгоритму при паралельному виконанні становить $O(log^2 n)$ кроків, що значно швидше за послідовні алгоритми зі складністю $O(n \log n)$.

Програмна реалізація виконана мовою C++ з використанням технології CUDA [4, 5]. Процес сортування розділений на декілька етапів (stages) та кроків (steps), де на кожному кроці GPU-ядро (kernel) виконує масово-паралельні операції порівняння та перестановки (compare-and-swap). Реалізація базується на моделі обчислень SIMT, де тисячі потоків одночасно обробляють різні пари елементів масиву.

Основні аспекти реалізації включають:

- управління потоками - кожен потік обчислює індекси двох елементів для порівняння на основі свого унікального ідентифікатора (threadIdx та blockIdx); це дозволяє уникнути конфліктів доступу до пам'яті та забезпечити максимальне завантаження обчислювальних ядер відеокарти;
- обробка даних - для коректної роботи алгоритму розмір вхідного масиву має бути степенем двійки; у роботі реалізовано механізм передачі даних між оперативною пам’яттю (Host) та відеопам’яттю (Device) за допомогою функцій cudaMalloc та cudaMemcpy;
- оптимізація - у ході розробки було враховано ієрархію пам'яті CUDA; використання глобальної пам'яті для великих масивів дозволяє обробляти об'єми даних, що перевищують 16 мільйонів елементів ($2^{24}$), забезпечуючи при цьому стабільну швидкість обчислень.

Експериментальні дослідження показали, що розроблена паралельна програма демонструє високу продуктивність: сортування масиву з $2^{20}$ елементів займає лише 0,126 секунди, що підтверджує ефективність обраного підходу для задач Big Data.

## Висновки

Досліджено бітонічне сортування, як одну із ключових операцій у паралельній обробці даних, визначено основні сфери її застосування. Розглянуто принципи функціонування сортувальних мереж, теоретичні засади алгоритму Bitonic Sort, процеси побудови бітонічних послідовностей та їх подальшого злиття, описано алгоритм вирішення задачі та його математичне обґрунтування. На основі літературних джерел досліджено технологію CUDA, як програмно-апаратну архітектуру для паралельних обчислень, розглянуто принципи роботи графічних процесорів (модель SIMT) та на основі цих досліджень сформульовано ідею реалізації паралельного алгоритму сортування для масово-паралельних систем.

Програмно реалізовано задану задачу мовою C++ з використанням CUDA Toolkit та описано всі компоненти коду, включаючи розробку обчислювальних ядер (kernels). Також наведено блок-схеми та діаграми діяльності, що ілюструють взаємодію між Host та Device. Проведено тестування розробленої програми на масивах даних різної розмірності, проаналізовано її результати, досліджено час виконання та показники швидкодії.

Визначено, що застосування паралельного алгоритму бітонічного сортування на базі CUDA забезпечує значне прискорення обробки даних порівняно з послідовними методами, особливо на великих масивах елементів, що підтверджує ефективність використання графічних процесорів для задач сортування.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1. Bitonic Sort. URL: https://www.baeldung.com/cs/bitonic-sort
2. Internals of Bitonic Sort. URL: https://xilinx.github.io/Vitis_Libraries/database/2022.1/guide/sort/bitonic_sort.html
3. Bitonic Sort with CUDA. URL: https://forums.developer.nvidia.com/t/bitonic-sort-with-cuda/34246
4. NVIDIA. CUDA C++ Programming Guide. URL: https://docs.nvidia.com/cuda/cuda-c-programming-guide/
5. Microsoft. C++ documentation. Learning resources and reference for modern C++. Microsoft Learn, 2024. URL:: https://learn.microsoft.com/en-us/cpp/

***Гурський Даніл Валентинович*** – студент кафедри комп`ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e- mail: danilahurskiy@gmail.com;

***Денисюк Валерій Олександрович*** – канд. техн. наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, e-mail: vad64@i.ua.

***Hurskyi Danil Valentinovych*** – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: danilahurskiy@gmail.com;

***Denysiuk Valerii Olexandrovich*** – Ph.D., Assistant Professor, Assistant Professor of the Chair of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: vad64@i.ua