

А.О. Гриценко, С.Ю. Сіроштан, Ю.Є. Зінченко
Донецький національний технічний університет

У статті запропоновано структуру програмно-апаратного модулю сортування. Використання модулю дозволяє зменшити час сортування за рахунок застосування спеціалізованого апаратного блоку. Також запропоновано варіант архітектури апаратного блоку сортування, що є адаптований для інтеграції із сучасними мікропроцесорними шинами систем на кристалі, зокрема, шиною Avalon мікропроцесора Nios II.

Постановка задачі. Безпосередньо поняття сортування та найбільш відомі алгоритми для вирішення цієї задачі розглянуто у [1, 2]. Згідно них оптимізація процесу сортування є важливою задачею. На даний момент найширше використовуваним алгоритмом програмного сортування є quicksort, що був запропонований в [3]. Деталі реалізації алгоритму розглянуто в [4], а, пізніше, відповідно до реалізації у стандартній бібліотеці мови програмування С [5], у [6].

Згідно [1, 2, 3, 4, 6] алгоритм quicksort базується на поступовому розділенні вихідного масиву. При цьому, згідно, наприклад [6], коли розмір масиву стає достатньо малим, треба використовувати алгоритм сортування вставками, тому що алгоритм quicksort у даному випадку не є ефективним.

У статті пропонується замінити алгоритм сортування вставками на використання спеціалізованого апаратного модулю. Це дозволить використати відомі переваги алгоритму quicksort при обробці великих вихідних масивів, а також переваги використання апаратної реалізації при впорядкуванні невеликих вторинних масивів.

Архітектура апаратного модулю сортування. У [1, 2] надано опис архітектури мереж сортування, що базується на [7]. Нажаль, використання цієї архітектури для такої платформи як, наприклад, ПЛІС FPGA не ефективно через дуже велике використання ресурсів. Тому, в якості базової архітектури апаратного модулю, який виконує сортування було обрано варіант, розглянутий в [8].

Але ж цей варіант потребує адаптації до обраної платформи через те, що сучасні мікропроцесори, зокрема використовуваний Nios II [9], взаємодіє із периферійними пристроями за допомогою специфічного

шинного інтерфейсу. У випадку мікропроцесору Nios II це інтерфейс Avalon [10] (рис. 1).

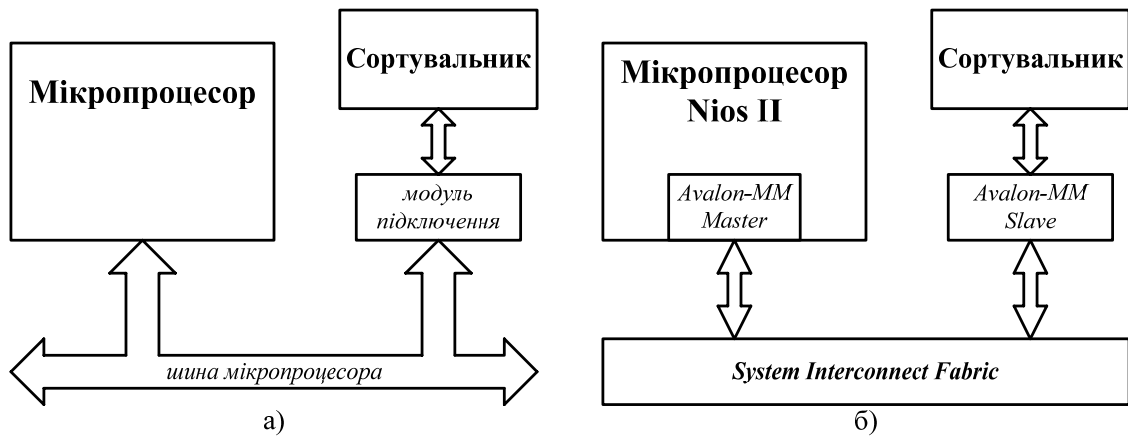


Рис. 1. Узагальнена та специфічна для Nios II схеми підключення модулю сортування

Відповідно до архітектури, наведеній у [8], запропонований варіант апаратного модулю сортування складається з декількох блоків, кожен з яких забезпечує впорядкування двох чисел (рис. 2).

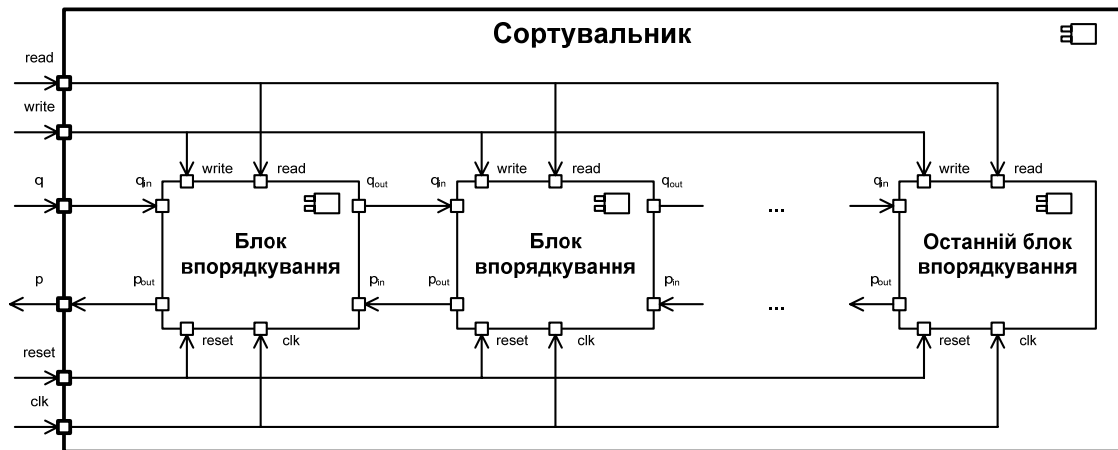


Рис. 2. Архітектура апаратного модулю сортування (діаграма компонентів у нотації UML [11])

Отож, для впорядкування масиву, що складається із N елементів, необхідно використати $N/2$ блоків. Слід зауважити, що останній блок може мати специфічну реалізацію, що дозволить зменшити апаратні затрати, але це не є обов'язковим.

На відміну від архітектури, показаної у [8] запропонований варіант не потребує окремого керуючого автомату. Керування здійснюється за

допомогою сигналів запису та зчитування, які є частиною інтерфейсу шини Avalon [10].

Окрім цього, пропонується архітектура, що підтримує синхронізацію з використанням відповідного сигналу, що також подається через шину Avalon.

Пропонуваний варіант має декілька переваг, по-перше, кількість блоків впорядкування визначає тільки максимально можливу кількість елементів масиву. При цьому модуль сортування, який має M блоків, може використовуватися для обробки масивів розміром від 1 до $M/2$, включно. По-друге, після ініціалізації системи цей блок не потребує додаткових циклів встановлення в початковий стан після завершення сортування масиву. Перехід в цей стан виконується автоматично після того, як з модулю прочитано впорядкований масив. По-третє, модуль побудовано таким чином, що він може бути легко налаштований для конкретних даних. Для цього визначено декілька точок налаштування, що показані на рис. 3.

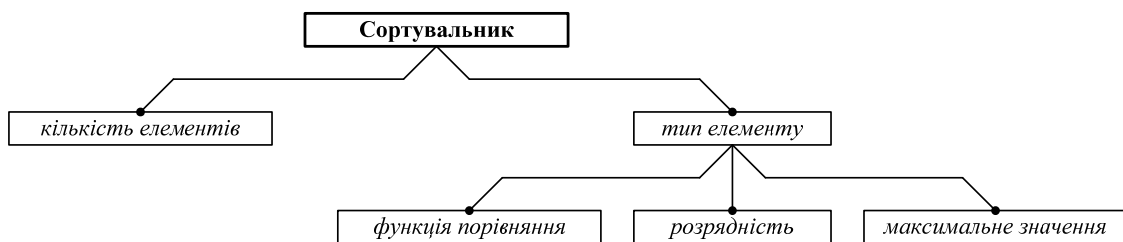


Рис. 3. Точки налаштування модулю сортування

Архітектура програмно-апаратного модулю. Отож, для створення програмно-апаратного модулю сортування необхідно поєднати якісну програмну реалізацію алгоритму quicksort та показаний вище модуль сортування на платформі Nios II (рис. 4).

В якості реалізації функції quicksort був обраний варіант, який є частиною стандартної бібліотеки GNU [12].

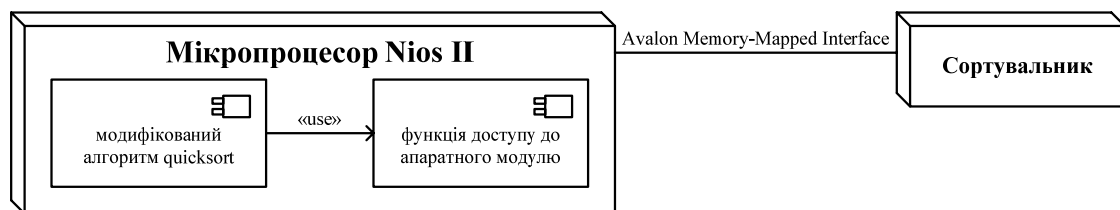


Рис. 4. Архітектура програмно-апаратного модулю (діаграма розгортання)

Інтерфейс обраної функції було розширено шляхом додання двох параметрів: функції, яка забезпечує взаємодію з модулем сортування, та розміром масиву, який можна впорядкувати апаратно.

Результати апробації. Апробацію запропонованого програмно-апаратного модулю було виконано з використанням ПЛІС FPGA Altera Cyclone III та відповідної мікропроцесорної архітектури Nios II з застосуванням комплекту Nios II Embedded Evaluation Kit. Отримані при цьому результати показано на рис. 5. Початковими даними експериментів були розмір вихідного масиву та максимальна ємність апаратного сортувальника.

Для порівняння було використано результати роботи алгоритму quicksort, який оптимізовано для мікропроцесора Nios II (є частиною стандартної бібліотеки C для цього мікропроцесору).

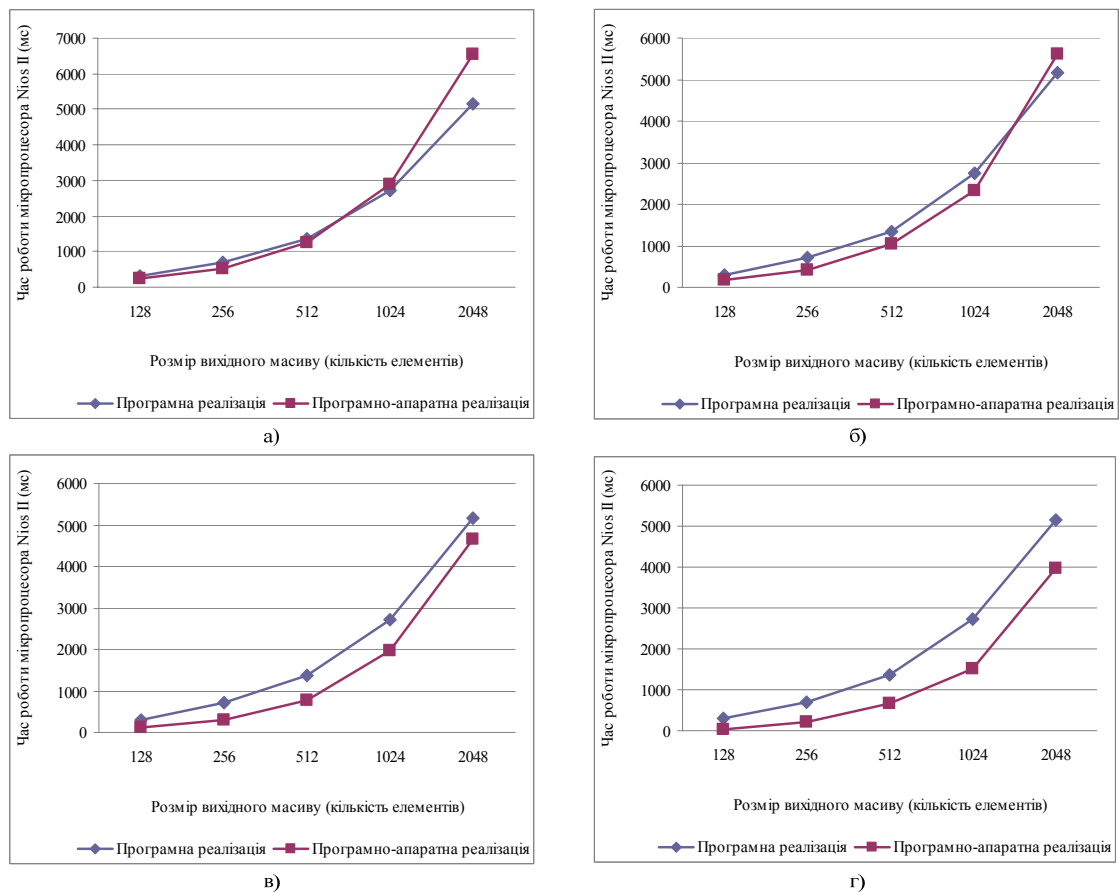


Рис. 5. Результати апробації програмно-апаратного модулю

Відповідно до наведених результатів можна визначити, по-перше, у тому разі, якщо розмір масиву значно перевищує ємність апаратного сортувальника, час роботи алгоритму може деградувати. Причиною є те, що час обміну даними з апаратним модулем перевищує час, який

необхідний для роботи програмної реалізації алгоритму сортування. По-друге, запропонована оптимізація гарантує приріст продуктивності на менш ніж на 20 відсотків, якщо розмір вихідного масиву більший за ємність апаратного модулю не більш ніж у 16 разів.

Висновки

У статті було запропоновано спосіб побудови апаратного модулю сортування. В свою чергу, на його основі з використанням алгоритму quicksort було побудовано програмно-апаратний модуль сортування. Цей модуль надає можливість пришвидшити процес сортування у разі наявності необхідних апаратних ресурсів.

Згідно до результатів експериментів, запропонована у даній статті реалізація модулю програмно-апаратного сортування є ефективною.

Бібліографічний список

1. Кнут Д. Искусство программирования, том 3. Сортировка и поиск / Кнут Д. – [2-е изд.] – М.: ООО «И.Д. Вильямс», 2007 – 832 с.: ил.
2. Алгоритмы: построение и анализ / [Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.] – [2-е изд.] – М.: Издательский дом «Вильямс», 2007. – 1296 с.: ил.
3. Hoare C.A.R. Quicksort / C.A.R. Hoare // Computer Journal. – 1962. – № 5 – P. 10-15
4. Sedgewick R. Implementing quicksort programs / R. Sedgewick // Communications of the ACM. – 1978 – № 21 – P. 847-857
5. Керниган Б. Язык программирования С / Б. Керниган, Д. Ритчи – [3-е изд.] – СПб.: «Невский диалект», 2001. – 352 с.: ил.
6. Bentley J.L. Engineering a Sort Function / J.L. Bentley, M.D. McIlroy // Software – Practice And Experience. – 1993 – № 23 – P. 1249-1265
7. Sorting networks and their applications / Batcher K.E. // Spring Joint Computer Conference. – 1968 – P. 307-314
8. Палагин А.В. Реконфигурируемые вычислительные системы: Основы и приложения / А.В. Палагин, В.Н. Опанасенко – К.: Просвіта, 2006. – 280 с.: ил.
9. Nios II Processor Reference Handbook [Electronic Resource]. – Mode of access : URL http://www.altera.com/literature/hb/nios2/n2cpu_nii5v1.pdf – Title from the screen.
10. Avalon Interface Specifications [Electronic Resource] – Mode of access : URL http://www.altera.com/literature/manual/mnl_avalon_spec.pdf – Title from the screen.
11. Объектно-ориентированный анализ и проектирование с примерами приложений / [Г. Буч, Р. Максимчук, М. Энгл и др.] – [3-е изд.] – М. : ООО "И.Д. Вильямс", 2008. – 720 с.: ил.
12. Index of /gnu/glibc [Electronic Resource] – Mode of access : URL <http://ftp.gnu.org/gnu/glibc/> – Title from the screen.