

УДК 004.31

А.О.Мельник, М.Аль Хабабсах

Національний університет „Львівська політехніка”

СТРУКТУРНА ОРГАНІЗАЦІЯ ЛАНОК ПОТОКОВИХ ПРОЦЕСОРІВ НА ОСНОВІ ПАМ'ЯТІ З ВПОРЯДКОВАНИМ ДОСТУПОМ

В роботі розглянуті питання побудови поточкових графів алгоритмів та їх реалізації в поточкових процесорах. Описана структура поточкових процесорів. Запропоновано використовувати пам'ять з впорядкованим доступом в ланках поточкових процесорів для виконання функції зберігання та впорядкування даних для їх подачі на входи операційного пристрою. Запропоновано базові структури ланок поточкових процесорів на основі пам'яті з впорядкованим доступом – лінійну та ітераційну. Наведено принципи роботи названих базових структур ланок поточкових процесорів. Запропоновано проводити модифікацію ПГА для підвищення ефективності поточкового процесора.

Ключові слова: *структурна організація ланок поточкових процесорів, поточкові графи алгоритмів, поточкові процесори, пам'ять з впорядкованим доступом.*

Вступ. Спеціалізовані процесори створюють для реалізації одного або декількох заданих алгоритмів [1]. Вони займають значну частину ринку комп'ютерних засобів, тому питання дослідження принципів їх побудови постійно знаходиться в сфері уваги розробників комп'ютерів. Зазвичай спеціалізовані процесори повинні обробляти інтенсивні потоки даних в темпі їх поступлення, тобто в реальному масштабі часу, причому з проведенням обробки раніше прийнятого масиву даних одночасно з прийомом нового масиву даних. Найкраще з цією задачею справляються процесори з поточною структурою (поточкові процесори), які складаються з послідовно з'єднаних процесорних ланок (ПБ), в кожній з яких виконуються операції одного ярусу поточкового графа алгоритму [2], який реалізується в процесорі. В даній роботі розглядаються питання структурної організації ланок поточкових процесорів.

Огляд літератури. Питанню структурної організації спеціалізованих процесорів присвячена значна кількість робіт [3-7]. Одним з видів цих процесорів є поточкові процесори, які особливо ефективні при виконанні алгоритмів опрацювання потоків даних в реальному масштабі часу, до яких, зокрема, належать алгоритми цифрового опрацювання сигналів та зображень [8]. Поточкові процесори складаються з послідовно з'єднаних процесорних ланок, в кожній з яких виконуються операції одного ярусу поточкового графа алгоритму [2], який реалізується в процесорі. Між процесорними ланками відповідно до структури алгоритму здійснюється інтенсивний обмін інформацією, причому потік є одностороннім: від ланок, які виконують операції початкових ярусів графа алгоритму, до ланок, які виконують операції кінцевих ярусів графа алгоритму. У зв'язку з цим значне навантаження лягає на пам'ять процесорних ланок, яка повинна бути паралельною, забезпечувати можливість одночасного безконфліктного доступу до даних з багатьох портів, одночасний запис даних, які поступають на її входи, та зчитування на виходи раніше записаних даних, можливість виконання операцій реорганізації масивів та впорядкування даних в масивах [2]. Як показано в ряді робіт [9-10], в ролі такої пам'яті в поточкових процесорах доцільно використати пам'ять з впорядкованим доступом (ПВД), яка задовольняє вищевказані вимоги та не вимагає звернення як при записуванні даних, так і при їх зчитуванні, звернення до комірки пам'яті і при записуванні, і при зчитуванні числа, та виключає потребу зберігання інформації про місце знаходження даного в пам'яті.

Постановка задачі. Питання використання пам'яті з впорядкованим доступом при побудові ланок поточкових процесорів до цих пір не розглядалось. Тому актуальним є дослідження принципів структурної організації ланок поточкових процесорів на основі пам'яті з впорядкованим доступом, яка має іншу організацію, аніж пам'ять з довільним доступом, яку зазвичай використовують при побудові спеціалізованих процесорів.

Поточковий граф алгоритму. Граф є однією з можливих форм подання алгоритму. Правило побудови графа алгоритму (ГА): функціональні оператори позначають кружками, а їх зв'язок – дугами. Вершина, яка відповідає функціональному оператору F_i ($i = \overline{1, N}$, де N – загальна

кількість вершин ГА), з'єднується з вершиною, яка відповідає функціональному оператору F_j ($j = \overline{1, N}$, причому $j > i$), тільки в тому випадку, коли одержаний після виконання оператора F_i результат є одним із аргументів для оператора F_j .

Вважають, що всі вхідні дані переміщуються дугами графа одночасно. Якщо на шляху даних, які переміщуються дугами, зустрічається вершина, то дані беруть участь в операції, заданій вершиною графа. В результаті перетікання даних між вершинами графа і виконання операцій, заданих алгоритмом, на виході дуги, яка виходить із вершини графа, що відповідає останній операції алгоритму, міститься результат виконання алгоритму.

Щоб оцінити обчислювальні й структурні характеристики алгоритму, його подають у вигляді потокового графа (ПГА) [2]. Потоковий граф алгоритму має N вершин, з'єднаних дугами, де кількість вхідних дуг будь-якої вершини графа k ($k = \overline{1, N}$) дорівнює a ($a = \overline{1, n}$), а вихідних дуг будь-якої вершини k дорівнює b ($b = \overline{1, n}$), n – кількість вхідних дуг графа, вершини нумерують натуральними числами $k = \overline{1, N}$ і розподіляють за l ярусами таким чином, що в i -му ярусі ($i = \overline{0, l-1}$) розміщені тільки вершини, на які надходять дуги від вершин попередніх ярусів і не надходять від вершин того ж ярусу та наступних ярусів, нумерацію ярусів починають з ярусу, на якому виконуються операції над вхідними даними, і закінчують ярусом, в якому знаходяться вихідні дуги графа, а нумерацію дуг розпочинають з першого ярусу і кожній вхідній дузі даного ярусу привласнюють натуральне число j ($j = \overline{1, n}$), а від нього вниз за правилом: вихідним дугам привласнюють довільне значення із множини натуральних чисел, що привласнені дугам, які входять в дану вершину.

В межах одного ярусу між вершинами потокового графа зв'язків немає. Вважають, що всі вхідні дані переміщуються дугами ПГА одночасно. Якщо на шляху даних, які переміщуються дугами, зустрічається вершина, то дані беруть участь в операції, заданій вершиною ПГА - ФО, інакше – просто передаються на наступний ярус. В результаті перетікання даних між ФО ПГА і виконання операцій, заданих алгоритмом, на виході ПГА отримують результат виконання алгоритму. В якості прикладу на рис.1 подано графічне зображення алгоритму обчислення виразу $y = (a + b) \times (c - d) / (e + f)$.

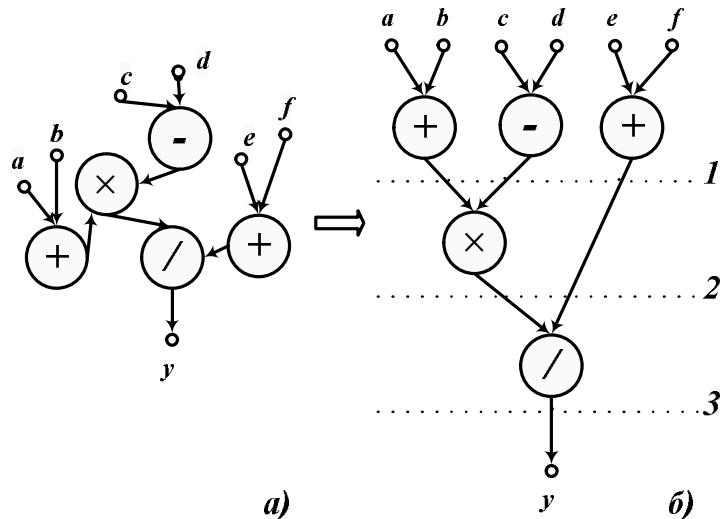


Рис. 1. а – граф алгоритму; б – потоковий граф алгоритму

Така форма подання алгоритму відображає, крім множини N вершин та множини M дуг ПГА, також степінь паралелізму графа - максимальну кількість вершин w в найширшому ярусі та мінімально можливий час обчислення даного алгоритму - кількість ярусів l , тобто ПГА є наочним представленням алгоритму і відображає його просторово-часові та обчислювальні характеристики.

Ланки поточкових процесорів. Потоковий процесор складається з послідовно з'єднаних ланок, в кожній з яких апаратно реалізується один ярус потокового графа алгоритму [2] Заданий алгоритм виконується над вхідними даними в разі їх одноразового проходження через всі ланки

потокowego процесора. Налаштування частоти роботи конвеєра на частоту надходження вхідних даних забезпечує неперервну обробку цих даних у реальному часі. Найбільша довжина конвеєра, тобто кількість ланок, визначається кількістю ярусів потокowego графа алгоритму.

Залежно від потрібної продуктивності в потокowym процесорі може апаратно відображатися як ПГА, так і його проекція. Використання апаратного відображення проекції потокowego графа алгоритму передбачає формування не тільки проекції функціональних операторів ярусу потокowego графа, а й проекції каналів передавання даних. Підхід апаратного відображення проекції функціональних операторів i -го ярусу (рис.2) є гнучкішим порівняно з апаратним відображенням ПГА [2].

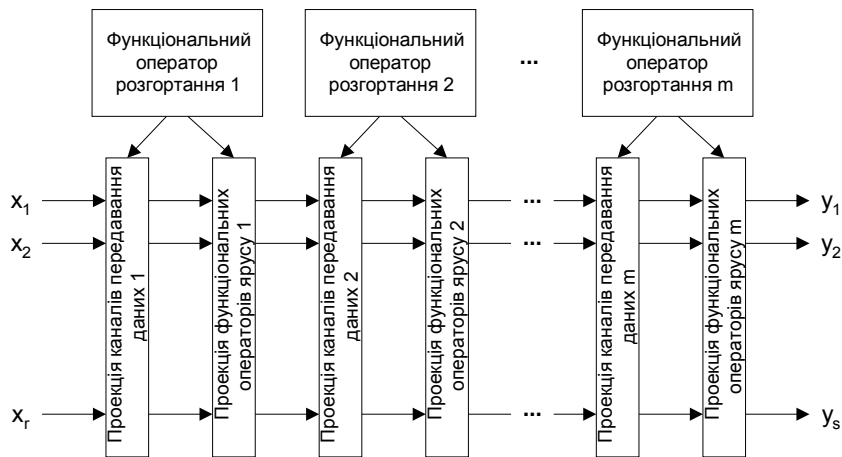


Рис. 2. Проекція потокowego графа алгоритму

Проекція функціональних операторів i -го ярусу є макрооперацією, на базі якої можуть бути виконані всі функціональні оператори цього ярусу. Проекція каналів передавання даних i -го ярусу є функціональним оператором затримки та перестановки даних для їх передавання у проекцію функціональних операторів ярусу. Отже, отриманий граф проекції функціональних операторів ярусів алгоритму є ущільненим у часі потокowym графом алгоритму. Для його відновлення в процесі обробки даних треба ввести функціональних оператор розгортання, який зберігає інформацію про структуру потокowego графа алгоритму. Кожному ярусу проекції потокowego графа алгоритму можна поставити у відповідність процесорну ланку, яка виконує операції цього ярусу. У складі такої процесорної ланки будуть буферна пам'ять, операційний пристрій і пристрій керування, що виконують, відповідно, операції проекції каналів передавання даних, проекції функціональних операторів ярусів та функціональні оператори розгортання.

Пам'ять з впорядкованим доступом. Як і пам'ять з послідовним доступом, ПВД орієнтована на роботу з масивами даних. В цій пам'яті забезпечується доступ до даних у програмно встановленому порядку, тобто мітка, яка поступає в пам'ять разом з даним або під час його зчитування, вказує місце даного у вихідному масиві.

В пам'яті з впорядкованим доступом відсутня прив'язка числа до конкретної комірки пам'яті, як це є в пам'яті з довільним доступом, а також відсутня ознака, за якою вона шукається, як це є в пам'яті з асоціативним доступом.

Робота пам'яті з впорядкованим доступом організована наступним чином [9,10]. Разом із даним в пам'ять поступає мітка, яка вміщує інформацію про подальше використання даного, а саме номер даного в масиві, за яким дане має з'явитись на виході пам'яті. При багатократному використанні того ж самого даного можливі використання відповідної кількості міток або повторний запис даного в пам'ять з новою міткою. Всередині пам'яті з впорядкованим доступом дані запам'ятовуються та проводяться дії щодо їх переміщення до виходу пам'яті з тим, щоб вони там з'явилися відповідно до величини їх мітки.

Враховуючи описані можливості пам'яті з впорядкованим доступом, не викликає сумніву доцільність її використання в якості буферної пам'яті ланки потокowego процесора.

Базові структури ланок потокowych процесорів на основі пам'яті з впорядкованим доступом. Можливі дві базові структури ланок потокowych процесорів на основі пам'яті з впорядкованим доступом: лінійна та рекурсивна.

До складу ланки потокowego процесора лінійної структури входить пам'ять з впорядкованим доступом, операційний пристрій і блок керування (рис.3).

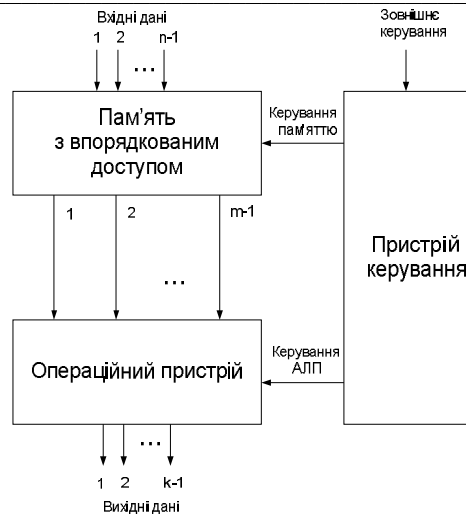


Рис. 3

В ланці потокового процесора лінійної структури відсутні обернені зв'язки з виходів операційного пристрою до пам'яті з впорядкованим доступом, яка зберігає вхідні дані ланки та виконує впорядкування вхідних даних для їх подачі на входи операційного пристрою. В операційному пристрої ланки лінійної структури виконуються всі функціональні оператори одного ярусу алгоритму за один прохід через нього масиву вхідних даних.

Крім синхронізації роботи операційного пристрою і пам'яті з впорядкованим доступом, блок керування формує коди виконуваних в операційному пристрої операцій, а також мітки, які супроводжують початкові дані в пам'яті з впорядкованим доступом. Ці мітки вказують порядок розміщення даних в вихідній матриці, яка зчитується з пам'яті з впорядкованим доступом.

В операційному пристрої ланки потокового процесора ітераційної структури (рис.4) виконуються функціональні оператори ярусу виконуваного алгоритму за кілька проходів через нього масиву початкових та проміжних даних.

Крім операційного пристрою, до складу ланки потокового процесора цього типу входять пам'ять з впорядкованим доступом і блок керування. В ланці потокового процесора присутні обернені зв'язки з виходів операційного пристрою до пам'яті з впорядкованим доступом, яка забезпечує зберігання та виконує впорядкування даних для їх подачі на входи операційного пристрою. Крім синхронізації роботи операційного пристрою і пам'яті з впорядкованим доступом, блок керування формує коди виконуваних в операційному пристрої операцій і мітки, які супроводжують початкові і проміжні дані в пам'яті з впорядкованим доступом. Ці мітки вказують порядок розміщення даних в вихідній матриці, яка зчитується з пам'яті з впорядкованим доступом. Операційні пристрої ланки потокового процесора цього типу в основному є багатофункціональними. Однофункціональними вони будуть тільки у випадку, коли апаратно-орієнтований процесор виконує алгоритм, всі функціональні оператори якого однакові. Тому блок керування також задає коди операцій операційного пристрою та організує виконання ітераційних обчислень.

Модифікація ПГА для підвищення ефективності потокового процесора. Оскільки в загальному випадку розміщення вершин за ярусами ПГА є нерівномірним, це призводить до різного навантаження на ланки потокового процесора. Для врівноваження цього навантаження може виникнути потреба в модифікації ПГА шляхом зміни ширини його ярусів. При цьому модифіковані ПГА є еквівалентними та відрізняються між собою лише топологією: від максимальної паралельної форми до послідовної. В якості прикладу на рис.5 наведено три варіанти ПГА алгоритму ШПФ для $N=8$ відповідно з чотирма, трьома та двома функціональними операторами в одному ярусі.

Для виконання побудови модифікованих ПГА та їх дослідження доцільно використати графічну систему ОСА [11]. Працюючи в цій системі, можна отримати інформацію про таблицю параметрів всіх можливих потокових графів того ж алгоритму з різною шириною, кількістю ярусів, мінімальною та максимальною кількістю ФО в ярусі. Дана інформація дозволяє вибрати ширину ПГА, при якій ланки потокового процесора є максимально завантаженими. Також можливе отримання модифікації ПГА із заданою шириною. Це дозволяє визначити кількість ярусів ПГА та розподіл ФО за ярусами.

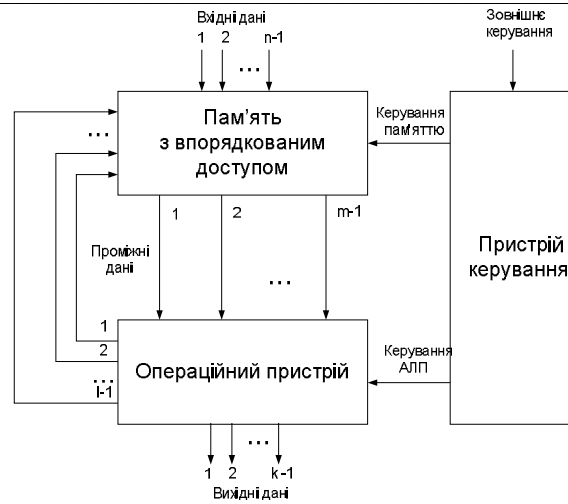


Рис. 4

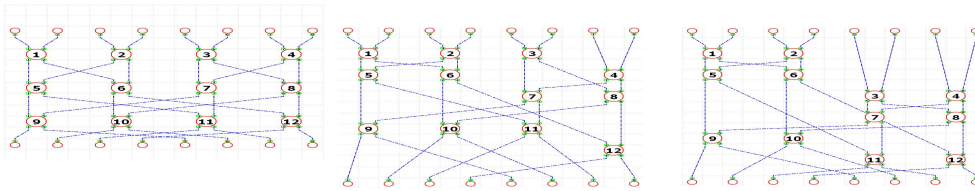


Рис. 6. Множина ПА ШПФ для N=8

Висновки

В роботі поставлена задача дослідження принципів структурної організації ланок потокових процесорів на основі пам'яті з впорядкованим доступом, яка має іншу організацію, аніж пам'ять з довільним доступом, яку зазвичай використовують при побудові спеціалізованих процесорів. Розглянуті питання побудови потокових графів алгоритмів та їх реалізації в потокових процесорах. Наведено принципи роботи названих базових структур ланок потокових процесорів. Запропоновано проводити модифікацію ПА для підвищення ефективності потокового процесора.

1. Кун С. Матричные процессоры на СБИС: Пер. с англ. - М.: Мир, 1991. - 672с.
2. Мельник А.О. Специализованные компьютерные системы реального czasu. - Львів: НУ „Львівська політехніка”, 1996. - 60 с.
3. А. Мельник, Д. Равашдех, М. Хабабсах. Вимоги до пам'яті спеціалізованих процесорів. Радіоелектронні і комп'ютерні системи. - 2009, С. 213-217.
4. Мельник А.О., Тарасенко В.П. Сучасні ситуативно-методологічні аспекти створення спеціалізованих комп'ютерних систем // Наукові вісті НТУУ “КПІ”. - Київ, 1997. - №1. - С. 18 - 21.
5. Anatoly Melnyk, Andriy Salo. Automatic generation of ASICS. // Proceedings of NASA/ESA Conference on Adaptive Hardware and Systems AHS-2007, Edinburgh, UK, 2007, pp. 311-317.
6. Мельник А.О. Програмовані процесори обробки сигналів. Львів: Видавництво Національного університету "Львівська політехніка" - 2000. - 65 с.
7. D.A. Buell, J.M. Arnold, and W.J. Kleinfelder, eds., Splash 2: FPGAs in a Custom Computing Machine, IEEE CS Press, 1996.
8. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. — М.: «Мир», 1978. — 848 с.
9. Мельник А.О. Принципи побудови буферної сортувальної пам'яті. // Вісник Державного університету “Львівська Політехніка” “Комп'ютерна інженерія та інформаційні технології”. - N307.- 1996. - С.65-71.
10. A. Melnyk, J. al Rawashdeh, M. al Hababsah. Organization and application of the programmable ordered access memory. CADSM-2009, 24-29 February, 2009, Polyana-Svalyava, Ukraine.
11. Мельник А.О., Яковлева І.Д. Графічна система для дослідження та опрацювання структури алгоритмів // Матеріали 4-ї міжнародної науково-технічної конференції ACSN-2009. - Львів, 2009, С. 67-71.