

## АПАРАТНІ ЗАСОБИ СОРТУВАННЯ ДАНИХ МЕТОДОМ ЗЛИТТЯ В РЕАЛЬНОМУ ЧАСІ

© Цмоць І. Г., Антонів В. Я., 2015

**Сформовано вимоги та вибрано принципи побудови апаратних засобів реального часу, розроблено узгоджені потокові графи алгоритмів сортування масивів даних методом злиття та синтезовано на їх основі апаратні засоби сортування даних з високою ефективністю використання обладнання.**

**Ключові слова:** сортування даних, метод злиття, апаратні засоби, потоковий граф, реальний час.

**The requirements for the real-time hardware development have been formed. The principles of such development have been selected. Consistent flow graphs for sorting algorithms by merging data sets have been created. The hardware for sorting data with the high efficiency of equipment usage has been synthesized.**

**Key words:** data sorting, fusion method, hardware, flow graph, real time.

### Вступ. Загальна постановка проблеми

Розвиток інформаційних технологій характеризується розширенням галузей застосування, в значній частині яких вимагається паралельне сортування масивів даних у реальному часі. Забезпечити таке сортування даних можливо спеціалізованими засобами, архітектура яких апаратно відображає структуру алгоритму сортування і орієнтована на реалізацію у вигляді надвеликих інтегральних схем (НВІС). Реалізація високоефективних спеціалізованих засобів сортування потребує широкого використання сучасної елементної бази, розроблення нових методів, алгоритмів і НВІС-структур. Режим реального часу та НВІС-реалізація алгоритмів сортування з високою ефективністю використання обладнання забезпечується розпаралелюванням і конвеєризацією процесів сортування, апаратним відображенням структури алгоритмів у архітектуру, яка адаптована до інтенсивності надходження потоків даних. Орієнтація структури засобів сортування на НВІС-реалізацію вимагає зменшення кількості виводів інтерфейсу та реалізацію алгоритмів на базі однотипних процесорних елементів (ПЕ) з регулярними та локальними зв'язками. У зв'язку з цим особливої актуальності набуває проблема розробки нових ефективних методів і алгоритмів сортування масивів даних, орієнтованих на адаптацію до інтенсивності надходження потоків даних і на НВІС-реалізацію.

### Аналіз останніх досліджень та публікацій

Аналіз методів паралельного сортування масивів даних показує [1–13], що для апаратної реалізації найбільше підходять методи сортування чисел підрахунком, витісненням, вставкою та злиттям. Порівняння цих методів [8–10] показує, що алгоритми реалізації методу паралельного сортування даних злиттям, порівняно з алгоритмами, які реалізують інші методи, є структурованішими, одноріднішими і орієнтованішими на паралельно-конвеєрну реалізацію у вигляді НВІС. В основі алгоритмів сортування методом злиття лежить базова операція об'єднання двох або більше упорядкованих масивів у один упорядкований масив. Апаратна реалізація базової операції об'єднання трьох і більше упорядкованих масивів в один упорядкований є складною і вимагає значних апаратних витрат. Простішою є базова операція об'єднання двох упорядкованих масивів в один упорядкований масив, тобто двошляхове злиття. Недоліком існуючих алгоритмів

реалізації двошляхового злиття є мала швидкодія, оскільки всі вони ґрунтуються на операціях попарного порівняння елементів даних [8,15,17]. Одним із шляхів підвищення швидкодії виконання двошляхового злиття є реалізація його на основі базової операції багатоканального злиття та пересилання груп даних. Кількість одночасно виконуваних базових операцій визначає швидкодію засобів сортування даних, яка завдяки симетрії більшості алгоритмів сортування чисел, може бути різною. Залежно від структури алгоритму, який реалізуються, та вимог конкретного застосування можуть бути синтезовані різні структури пристроїв сортування даних, які відрізняються і організацією процесу сортування, і за технічними параметрами [5].

З аналізу публікацій [1–18] випливає, що синтез з високоефективних апаратних засобів сортування даних у реальному часі методом злиття вимагає розроблення нових узгоджено-паралельних алгоритмів та методів просторово-часового перетворення їх у паралельні структури.

### Формування мети

**Метою роботи** є вибір принципів і формування вимог побудови до апаратних засобів сортування в реальному часі, розроблення узгоджених потокових графів алгоритму сортування масивів даних методом злиття та синтез на їх основі апаратних засобів з високою ефективністю використання обладнання.

### Аналіз отриманих наукових результатів

**Вибір принципів і формування вимог до побудови апаратних засобів сортування масивів даних.** Однією з найширше розповсюджених вимог, що ставиться до засобів паралельного сортування масивів даних є забезпечення високої швидкодії. Подібна проблема виникає, як правило, при використанні таких засобів для сортування інтенсивних потоків масивів даних у реальному часі. Для забезпечення реального часу необхідно, щоб час сортування  $T_c$  не перевищував час надходження даних  $T_{нд}$ , тобто:

$$T_c \leq T_{нд}.$$

Крім того, такі засоби повинні мати високу ефективність використання обладнання, яка враховує кількість виводів інтерфейсу, зв'язує продуктивність з витратами обладнання та дає оцінку елементам (вентиліям) за продуктивністю. Кількісна величина ефективності використання обладнання  $E$  при НВІС-реалізації алгоритмів сортування даних визначається так:

$$E = \frac{R}{t_c \left( \sum_{i=1}^H W_{ПЕ_i} + k_1 Y + k_2 P \right)},$$

де  $R$  – складність алгоритмів сортування даних, яка визначається кількістю операцій попарного порівняння;  $W_{ПЕ}$  – витрати обладнання на реалізацію ПЕ;  $H$  – кількість ПЕ;  $t_c$  – час сортування масиву даних;  $Y$  – кількість виводів інтерфейсу;  $k_1$  – коефіцієнт врахування кількості виводів інтерфейсу  $k_1=f(Y)$ ;  $P$  – кількість зв'язків між ПЕ;  $k_2$  – коефіцієнт врахування зв'язків між ПЕ  $k_2=f(P)$ .

Однією з умов досягнення високої ефективності використання обладнання під час сортування масивів даних у реальному часі є виконання такої умови:

$$P_d \leq D_c,$$

де  $P_d = k n_k F_d$  – інтенсивність надходження вхідних даних;  $D_c = \frac{s n_s}{T_k}$  – інтенсивність сортування;

$k$  – кількість каналів надходження вхідних даних;  $n_k$  – розрядність каналів надходження даних;  $F_d$  – частота надходження даних;  $s$  – кількість каналів сортування даних;  $n_s$  – розрядність каналів сортування даних;  $T_k$  – конвеєрний такт роботи пристрою сортування.

Для НВІС-реалізацій алгоритми сортування повинні бути добре структурованими, орієнтованими на реалізацію на множині взаємозв'язаних ПЕ та забезпечувати детерміноване переміщення даних. Структура та операції, які виконують ПЕ, залежить від вимог, що висуваються до часу сортування. При розробці алгоритмів сортування для НВІС-реалізацій потрібно одночасно

враховувати багато взаємопов'язаних факторів. Передусім необхідно, щоб алгоритми сортування були рекурсивними та локально залежними. У рекурсивному алгоритмі всі ПЕ повинні виконувати однакові операції.

Алгоритми сортування даних можна розділити на два класи з локальними та глобальними зв'язками між ПЕ. Для оцінювання зв'язків між ПЕ, часової та апаратної складності реалізації алгоритму розробляють решіткову модель, яка складається з множини ПЕ, що в декартовій системі координат утворюють точкові системи (решітки) [5]. У решітковій моделі кожному ПЕ ставиться у відповідність  $j$  часовий та просторовий  $i$  індекси, які вказують коли та де виконується кожна із операцій алгоритму. В алгоритмах з локальними пересилками даних різниця між просторовими індексами  $j$  на кроці рекурсії обмежена деякою константою, оскільки в таких алгоритмах обміни здійснюються тільки між найближчими сусідніми ПЕ. Алгоритми, що при рекурсії мають рознесені просторові індекси, відносяться до класу алгоритмів з глобальними зв'язками. Складність реалізації обміну між ПЕ залежить і від розрядності каналів передавання даних, і від віддалі  $\Delta i$  між ними, яка визначається як різниця двох просторових індексів.

Вартість НВІС для паралельного сортування масивів даних в основному залежить від площі кристала, яка визначається і витратами обладнання (кількість транзисторів), і кількістю зовнішніх виводів, число яких обмежене рівнем технології та розміром кристала. Орієнтація структур сортування даних на НВІС-реалізацію вимагає зменшення числа виводів інтерфейсу та кількості з'єднань між ПЕ.

Розробку високоефективних засобів паралельного сортування масивів даних можна забезпечити при інтегрованому підході, який охоплює:

- дослідження, розробку методів і алгоритмів паралельного сортування масивів даних великої розрядності;
- розроблення нових структурних і схемотехнічних рішень, орієнтованих на НВІС-технології;
- засоби автоматизованого проектування НВІС, які забезпечать зменшення термінів і підвищать якість проектування.

Для забезпечення високої ефективності використання обладнання при розробленні НВІС-структур для сортування масивів даних у реальному часі пропонується використовувати такі принципи [5,6]:

- розпаралелення процесу сортування даних;
- спеціалізації та адаптації апаратних засобів до структури алгоритмів сортування та інтенсивності надходження даних;
- однорідності ПЕ та регулярності зв'язків між ними;
- узгодженості інтенсивності сортування з інтенсивністю надходження даних.

**Форми відображення алгоритмів сортування.** Для оцінки обчислювальних і структурних характеристик алгоритмів сортування даних використовується їх подання у вигляді функціонального графа  $F=(\Phi, \Gamma)$ , де  $\Phi=\{\Phi_1, \Phi_2, \dots, \Phi_n\}$  – множина функціональних операторів,  $\Gamma$  – закон відображення зв'язків між операторами [5]. Функціональний граф алгоритму відображає послідовність і взаємну залежність функціональних операторів. Графічно функціональний граф алгоритму відображається у вигляді вершин, що відповідають операторам алгоритму  $\Phi_i$  та дуг, які відображають зв'язки між операторами. Складність функціональних операторів  $\Phi_i$  визначається і структурними одиницями інформації, і складністю виконуваних операцій. Таке подання алгоритму не повною мірою відображає просторово-часові залежності між функціональними операторами.

Виявити паралелізм алгоритму сортування даних, управляти ним для знаходження оптимальних просторово-часових рішень забезпечує подання функціонального графа в ярусно-паралельній формі (ЯПФ) [5]. При представленні у такій формі алгоритму відбувається розподіл всіх його функціональних операторів  $\Phi_i$  за ярусами так, що в  $j$ -му ярусі розміщені функціональні оператори, які залежать хоча б від одного функціонального оператора  $(j-1)$ -го ярусу і не залежать від операторів наступних ярусів. У середині ярусу між усіма функціональними операторами не має з'єднань.

Кожний  $j$ -й ярус алгоритму описується такими параметрами:

- наборами незалежних функціональних операторів  $\Phi_{ji}$ , де  $j$  – номер ярусу,  $i$  – номер функціонального оператора в ярусі;
- набором каналів надходження даних і видавання проміжних результатів;
- розрядністю кожного каналу зв'язку.

Кількість ярусів у ЯПФ алгоритму є її висотою  $h$ , а максимальна кількість функціональних операторів у ярусі визначає її ширину  $L$ . За допомогою просторово-часових індексів у системі координат час-простір задається розміщення функціональних операторів. Таке відображення орієнтованого графа алгоритму будемо називати потоковою паралельною формою або потоковим графом [5,6]. Параметри потокового графа, такі як складність функціональних операторів  $\Phi_{ji}$ , ширина  $L$  і висота  $h$  є взаємно залежними, під цим розуміється, що зміна одного спричиняє зміну інших.

Для переходу від функціонального графа до відображення його у вигляді потокового графа необхідно його записати в вигляді матриці  $n \times n$ , де “1” відповідає наявності каналу зв'язку, “0” – відсутності зв'язку [5]. Матриця формується так, що для кожного функціонального оператора  $\Phi_{ji}$  джерела інформації формується рядок, який відображає його зв'язки з іншими функціональними операторами. Якщо позначити через вектори  $\vec{V}_{\Phi_1}, \dots, \vec{V}_{\Phi_n}$  відповідні стовпці матриці, то можна визначити результуючий вектор-стовпець:

$$\vec{V}_0 = \vec{V}_{\Phi_1} + \vec{V}_{\Phi_2} + \dots + \vec{V}_{\Phi_n}$$

У векторі  $\vec{V}_0$  визначаємо номери елементів, які дорівнюють нулю, наприклад, другий та п'ятий. За визначеними номерами знаходимо функціональні оператори, що не мають нащадків і відповідно утворюють нульовий ярус, в даному випадку  $\Phi_2$  і  $\Phi_5$ . Далі за формулою обчислюємо:

$$\vec{V}_1 = \vec{V}_0 - \vec{V}_{\Phi_2} - \vec{V}_{\Phi_5}.$$

Номери нульових елементів знаходимо у векторі  $\vec{V}_1$ , за якими визначаємо функціональні оператори, що будуть утворювати перший ярус. Таким саме способом обчислюємо всі наступні вектори та визначаємо для них функціональні оператори, що будуть утворювати наступні яруси.

Для переходу від алгоритмів сортування до НВІС-архітектур з високою ефективністю використання обладнання пропонується такі алгоритми подавати в вигляді узгодженого потокового графа.

**Розробка узгодженого потокового графа для алгоритмів сортування.** Процес розробки узгодженого потокового графа для алгоритмів сортування можна розбити на такі чотири етапи:

- 1) декомпозиція алгоритму розв'язання задачі;
- 2) проектування комунікацій (обмінів даними) між функціональними операторами;
- 3) укрупнення функціональних операторів;
- 4) планування процесу сортування.

На етапі декомпозиції алгоритм сортування масивів даних  $\Phi$  розбивається на функціональні оператори  $\Phi_{ji}$ , між якими установлюються зв'язки, що відповідають даному алгоритму сортування. Що більший степінь деталізації алгоритму одержується у процесі декомпозиції, то простіше і легше здійснити його адаптацію до вимог конкретного застосування. Здійснити декомпозицію можна за допомогою методів функціональної декомпозиції. Використання методу функціональної декомпозиції дозволяє отримати просторово-часове відображення структури алгоритму сортування масивів даних на основі елементарних операцій попарного порівняння та перестановки даних. Час і спосіб виконання таких операцій ПЕ є одними із основних параметрів при визначенні конвеєрного такту роботи  $T_k$  і розрядності каналів надходження даних  $n_k$  у НВІС-структурах сортування даних. На першому етапі розробки ми отримуємо граф схему алгоритму, де функціональні оператори  $\Phi_{ji}$  мають приблизно однаковий час виконання, а необхідна швидкодія визначає їхню складність.

На етапі проектування комунікацій необхідно визначити структуру та розрядність каналів обміну даними між функціональними операторами  $\Phi_{ji}$ . Для цього проводиться перехід від графа схеми алгоритму до потокового графа, в якому відбувається закріплення функціональних операторів  $\Phi_i$  за ярусами і просторово-часове розміщення. Структура зв'язків між функціональними

операторами  $\Phi_{ji}$  у потоковому графі сусідніх ярусів визначається кількістю каналів надходження даних і їх розрядністю. Приклад, потокового графа алгоритму паралельного сортування методом злиття масиву із 16-и чисел, наведений на рис. 1, де  $\Phi_{ji}$  – функціональний оператор попарного порівняння та перестановки даних.

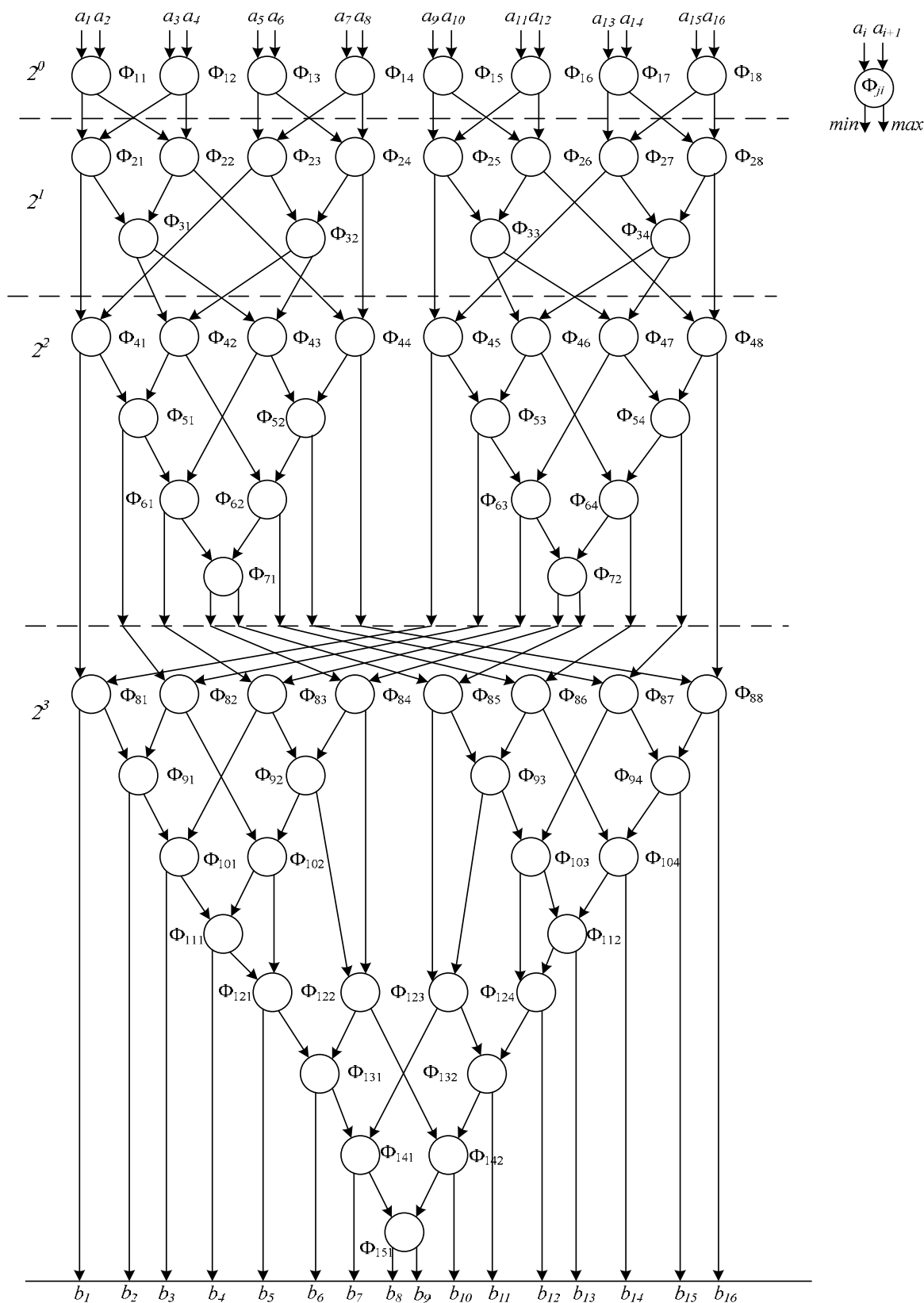


Рис. 1. Потоковий граф алгоритму паралельного сортування методом злиття масиву із 16-ти чисел

В основі алгоритмів сортування методом злиття лежить базова операція об'єднання двох упорядкованих масивів  $\{a_{1i}\}_{i=1}^{2^{i-1}}$  та  $\{a_{2i}\}_{i=1}^{2^{i-1}}$  в один упорядкований масив  $\{b_{1i}\}_{i=1}^{2^i}$ . На початку сортування вхідний масив чисел  $\{a_j\}_{j=1}^N$  розбивається на  $N/2$  упорядкованих масивів довжиною одиниця. У результаті виконання першої базової операції формуються  $N/4$  упорядкованих масивів довжиною два. Кількість типів базових операцій, необхідних для сортування масиву із  $N$  чисел визначається за формулою:

$$k = \log_2 N.$$

Алгоритм сортування масивів даних методом злиття реалізується на базі такого функціонального оператора  $\Phi_{ji}$ :

$$y = \begin{cases} 0, & \text{коли } a_1 \leq a_2 \\ 1, & \text{коли } a_1 > a_2 \end{cases}, \quad b_1 = \begin{cases} a_1, & \text{коли } y = 1 \\ a_2, & \text{коли } y = 0 \end{cases}, \quad b_2 = \begin{cases} a_1, & \text{коли } y = 0 \\ a_2, & \text{коли } y = 1 \end{cases},$$

де  $y$  – результат порівняння двох чисел;  $a_1, a_2$  – числа, які порівнюються;  $b_1, b_2$  – виходи більшого та меншого відсортованих чисел.

Потоковий граф алгоритму паралельного сортування методом злиття масиву із 16-ти чисел має висоту  $h=15$ , ширину  $L=8$  і реалізується на основі чотирьох типів базових операцій. У першому ярусі даного графа виконуються базової операції першого типу, у другому і третьому ярусах – базові операції другого типу, у четвертому, п'ятому, шостому і сьомому – базової операції третього типу, а у восьмому, ..., п'ятнадцятому – базової операції четвертого типу.

За результатами перших двох етапів розробки можна оцінити інтенсивність сортування масивів даних  $D_c$ , яку можна отримати при апаратній реалізації поточного графа сортування.

Вихідними даними для визначення інтенсивності сортування  $D_c = \frac{sn_s}{T_k}$  масивів даних  $\epsilon$ :

- кількість каналів сортування даних  $s$  і їх розрядність  $n_s$ ;
- складність функціональних операторів  $\Phi_{ji}$ , яка визначає час їх реалізації та конвеєрний такт сортування  $T_k$ ;
- швидкодія елементної бази, яка враховується під час визначення конвеєрного такту сортування  $T_k$ .

Для оцінки узгодженості інтенсивності надходження даних  $P_d$  із обчислювальної здатності  $D_c$  вводиться коефіцієнт узгодженості, який визначається так:

$$L = \left\lceil \frac{P_d}{D_c} \right\rceil$$

де  $\lceil \cdot \rceil$  – знак округлення до більшого цілого.

Коефіцієнт узгодженості  $L$  може бути  $L = 1$ ,  $L > 1$  та  $L < 1$ . Коли  $L = 1$ , то розроблений граф сортування даних  $\epsilon$  узгодженим і його  $\epsilon$  апаратна реалізація забезпечує високу ефективність використання обладнання.

Якщо  $L > 1$ , то розроблений граф сортування не  $\epsilon$  узгодженим і для його узгодження необхідно збільшувати інтенсивність сортування  $D_c$ . Підвищення інтенсивності сортування  $D_c$  може бути досягнуте шляхом збільшення кількості каналів сортування даних  $s$  і їх розрядності  $n_s$  або зменшення складності функціональних операторів  $\Phi_{ji}$ . Якщо зміною перерахованих параметрів не вдасться досягнути необхідної інтенсивності сортування  $D_c$ , то тоді підвищення інтенсивності сортування  $D_c$  досягається за рахунок паралельної реалізації  $L$  графів.

*Третій етап проектування*, який зводиться до укрупнення операцій шляхом об'єднання функціональних операторів  $\Phi_{jk}$  і каналів передавання даних і у межах ярусу, і між ярусами. Цей етап використовується для випадку  $L < 1$ , тобто коли необхідно зменшити інтенсивність сортування  $D_c$ . Граф сортування, який отримуємо у результаті такого об'єднання, будемо називати конкретизованим поточним графом. Етап укрупнення тісно пов'язаний з етапом планування процесу сортування.

Четвертий етап планування процесу сортування зводиться до збереження інформації про структуру потокового графа алгоритму сортування. На цьому етапі виконується планування процесу сортування, визначаються величини затримок і перестановки даних. Для відтворення процесу сортування даних у конкретизований потоковий граф вводяться оператори управління, затримки та перестановки даних. Розглянемо можливі три основні варіанти укрупнення операцій (об'єднання функціональних операторів) для отримання конкретизованого потокового графа сортування даних.

Першим із варіантів отримання конкретизованого графа сортування даних є його лінійна проекція на горизонтальну вісь  $X$ . У цьому випадку укрупнення операцій здійснюється об'єднанням між ярусами функціональних операторів і каналів передавання даних. Приклад проекції потокового графа алгоритму паралельного сортування методом злиття масиву із 16-ти чисел на горизонтальну вісь  $X$ , наведений на рис.2, де  $\Phi_{ji}$  – функціональний оператор попарного порівняння та перестановки даних,  $\Phi_{МЗП}$  – макрооператор затримки та перестановки,  $\Phi_{МУ}$  – макрооператор управління.

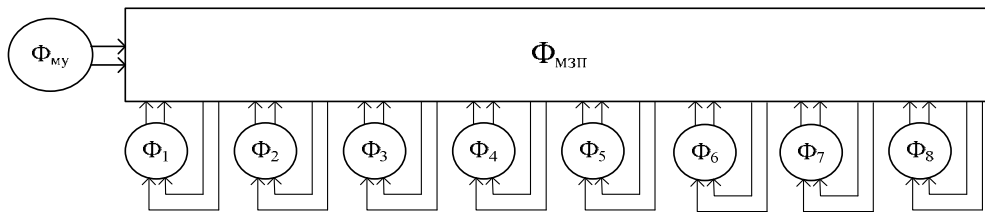


Рис. 2. Лінійна проекція графа сортування методом злиття масиву із 16-ти чисел на горизонтальну вісь  $X$

Оціночна інтенсивність сортування масивів даних для лінійної проекції на горизонтальну вісь дорівнює

$$D_c = \frac{sn_s}{(N-1)T_k},$$

де  $N$  – кількість чисел сортування.

Другим варіантом отримання конкретизованого потокового графа сортування даних методом злиття є його лінійна проекція на вертикальну вісь  $Y$ . У цьому випадку укрупнення операцій здійснюється об'єднанням функціональних операторів і каналів передавання даних і в межах ярусу, і між ярусами. У результаті такого укрупнення операцій отримуємо конкретизований потоковий граф з одним каналом передавання даних і чотирьох типів базових операцій злиття двох упорядкованих масивів розміром  $2^{i-1}$  в один упорядкований масив розміром  $2^i$ , де  $i=1, \dots, N$ . Приклад проекції потокового графа алгоритму паралельного сортування методом злиття масиву із 16-ти чисел на вертикальну вісь  $Y$  основі одноканального двошляхового злиття, наведений на рис.3а, де  $\Phi_{зп1} - \Phi_{зп4}$  – оператори затримки та перестановки для одноканального двошляхового злиття;  $\Phi_{у1} - \Phi_{у4}$  – оператори управління. Оціночна інтенсивність сортування масивів даних для лінійної проекції на вертикальну вісь  $Y$  дорівнює

$$D_c = \frac{n_s}{T_k}.$$

Третім варіантом отримання конкретизованого графа сортування даних методом злиття є використання для його реалізації базових операцій багатоканального двошляхового злиття. Кількість каналів двошляхового злиття залежить від коефіцієнта узгодженості  $L$ . Необхідна кількість базових операцій двошляхового паралельного злиття для сортування масиву з  $N$  даних, визначається за формулою:

$$g = \left\lceil \log_2 \frac{N}{m} \right\rceil,$$

де  $m$  – кількість каналів.

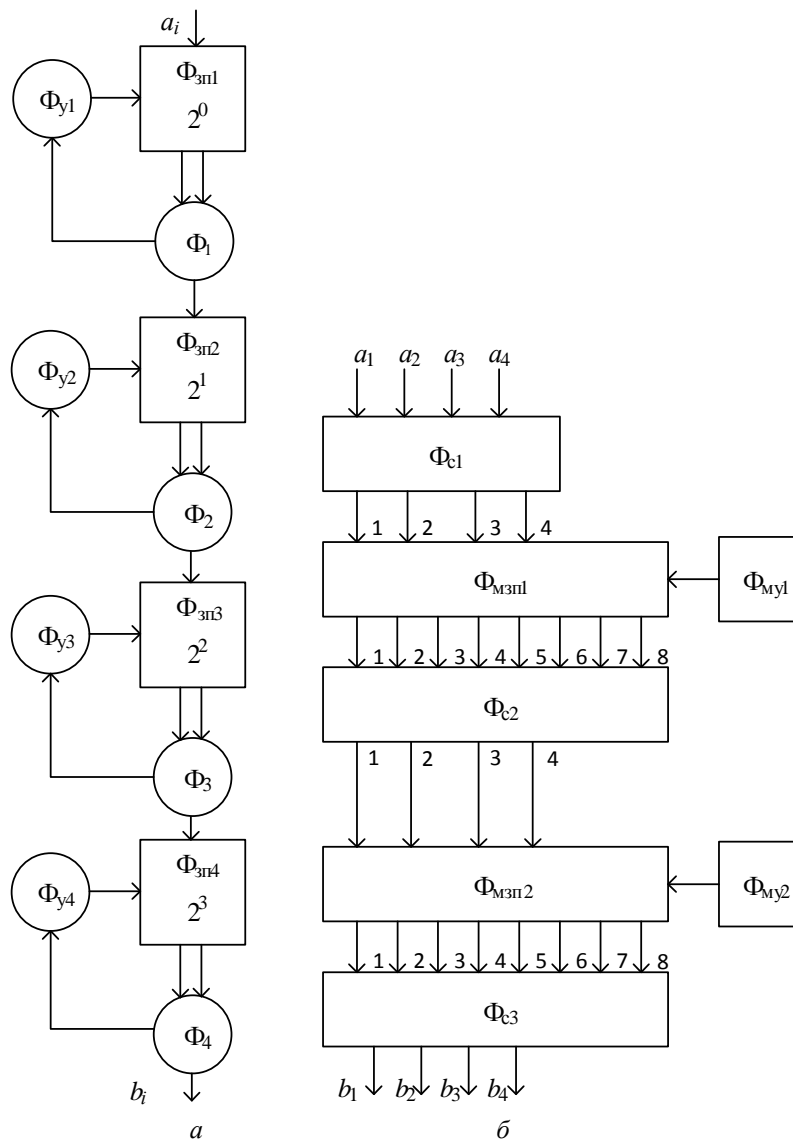


Рис. 3. Лінійна проекція на вертикальну вісь  $Y$  графа сортування масиву із 16-ти чисел, де  $a$  – на основі одноканального двошляхового злиття;  $b$  – на основі чотириканального двошляхового злиття

Проекція на вертикальну вісь  $Y$  графа сортування на основі багатоканального двошляхового злиття, наведена на рис.3б, де  $\Phi_{\text{мy}}$  – макрооператор управління,  $\Phi_{\text{с1}}$  – функціональний оператор сортування чотирьох чисел,  $\Phi_{\text{с2}}$ ,  $\Phi_{\text{с3}}$  – функціональні оператори сортування восьми чисел.

Оціночна інтенсивність сортування масивів даних на основі багатоканального двошляхового злиття для лінійної проекції на вертикальну вісь  $Y$  дорівнює

$$D_c = \frac{mn_s}{T_k}.$$

Основним шляхом підвищення інтенсивності сортування масивів даних на основі багатоканального двошляхового злиття є збільшення кількості каналів.

**Синтез апаратних засобів сортування масивів даних у реальному часі.** Для синтезу апаратних засобів сортування масивів даних у реального часу використовуємо відомий метод адекватного апаратного відображення структури графів алгоритмів. При використанні цього методу кожному функціональному оператору ставиться у відповідність ПЕ або блок сортування, макрооператору затримки та перестановки – пам'ять і комутатори, макрооператору управління – блок управління, а дугам між функціональними операторами – каналами передавання даних [5]. Синтезовані у такий спосіб апаратні засоби є алгоритмічними. У таких структурах алгоритм



реалізується при проходженні та обробці даних від входів до виходів через всі операційні блоки. За режимами роботи алгоритмічні структури діляться на синхронні та асинхронні.

В асинхронних (однотактних) структурах дані опрацюються без проміжних запам'ятовувань. Кожна однотактна структура є послідовною з погляду реалізації функціональних операторів Ф<sub>ji</sub>. Це спричинено обмеженою швидкістю та неефективністю використовуваного обладнання при обробці інтенсивних потоків даних у реальному часі. Використовування для обробки потоків даних синхронних структур з конвеєрною реалізацією графів алгоритмів є доцільним, оскільки відбувається суміщення у часі виконання функціональних операторів алгоритму над різними даними. Конвеєризація алгоритмічних структур передбачає розбиття їх на кроки шляхом введення буферної пам'яті. При цьому, кожен крок конвеєра буде складатися з двох компонентів: буферної пам'яті; ПЕ (блока сортування) блока управління, які реалізують оператори ярусу. Для забезпечення високої швидкості та ефективного використання обладнання функціональними операторами, які реалізуються у кроках конвеєра, мають бути простими та мати приблизно однаковий час реалізації. Однотактні алгоритмічні пристрої можна вважати одноступінчатим конвеєром. Зважаючи на це, актуальним постає розкриття питань, пов'язаних із синтезом конвеєрних алгоритмічних структур реального часу з високою ефективністю використання обладнання.

Вихідною інформацією для синтезу пристроїв сортування масивів даних реального часу є:

- кількість вхідних даних  $N$ ;
- розрядність вхідних даних;
- вимоги до інтерфейсу;
- інтенсивність надходження вхідних даних  $P_d = kn_k F_d$ ;
- техніко-економічні вимоги і обмеження.

Процес синтезу апаратних засобів сортування масивів даних у реальному часі є багаторівневим та ітераційним, з аналізом характеристик, з поверненням назад і переглядом раніше прийнятих рішень. Послідовна функціональна декомпозиція алгоритму сортування масивів даних з визначенням інтерфейсу апаратних засобів, функцій кожного функціонального оператора, операторів управління, затримки та перестановки відображає процес синтезу “зверху вниз”.

При синтезі пристроїв сортування масивів даних у реальному часі для досягнення високої ефективності використання обладнання необхідно забезпечити:

- узгодження інтенсивності надходження даних із інтенсивністю сортування;
- мінімізацію апаратних затрат при забезпеченні реального часу.

Основними шляхами зменшення апаратних затрат і узгодження інтенсивності надходження даних із інтенсивністю сортування є:

- вибір ефективних методів і алгоритмів сортування даних;
- вибір складності функціональних операторів;
- зміна розрядності каналів надходження даних і розрядності ПЕ;
- зміна кількості каналів надходження даних.

При синтезі матричного пристрою паралельного сортування методом злиття кожному функціональному оператору ставиться у відповідність ПЕ, які з'єднані між собою каналами передавання даних у відповідності до потокового графа алгоритму паралельного сортування (рис. 4). Структура можливих ПЕ, які реалізують функціональний оператор, наведена на рис. 4, де Тг – тригер, Км – комутатор, СП – схема порівняння, ТІ – вхід тактових імпульсів, R – вхід скиду в нуль.

У матричному пристрої паралельного сортування методом злиття одним із варіантів узгодження інтенсивності надходження даних із інтенсивністю сортування є зміна розрядності каналів надходження даних і розрядності ПЕ. На рис. 4 наведені два типи ПЕ:

- з паралельним надходженням і порівнянням даних (рис. 4, а);
- з порозрядним надходженням і порівнянням даних (рис. 4, б).

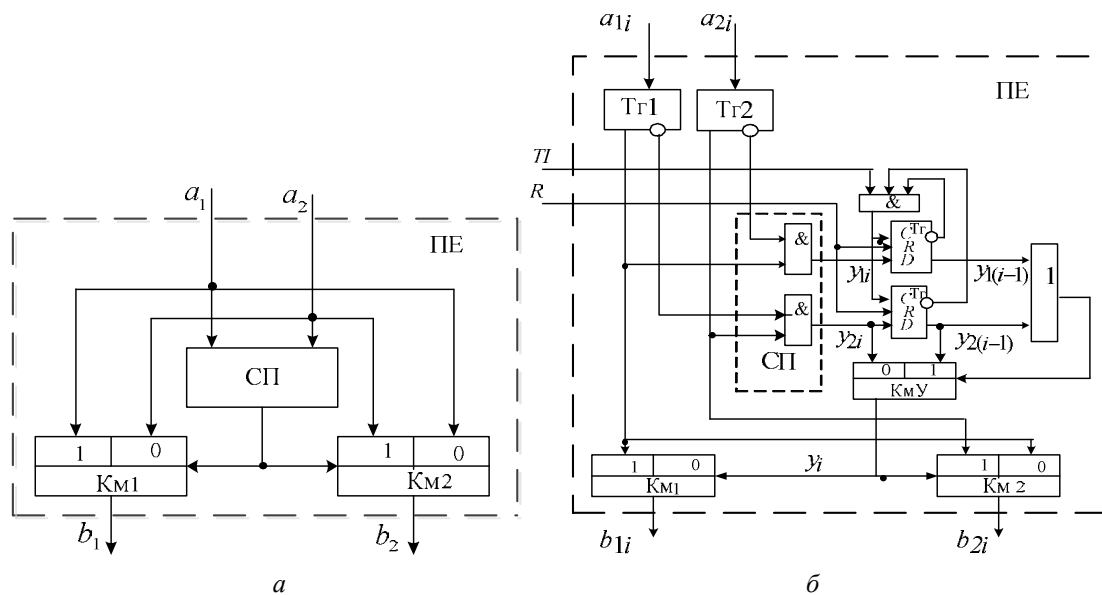


Рис. 4. Структура ПЕ:

*a* – з паралельним надходженням всіх розрядів даних; *б* – з порозрядним надходженням даних

Ці типи ПЕ є крайніми варіантами використання вертикально-групового надходження і порівнянням даних. У ПЕ (рис. 4, *a*) числа надходять паралельним кодом і їх порівняння здійснюється за один такт, а у ПЕ (рис. 4, *б*) числа надходять порозрядно і для їх порівняння необхідно  $n$  тактів. Формування сигналів порозрядного порівняння  $y_{1i}$  і  $y_{2i}$  у ПЕ (рис. 5, *б*) здійснюється СП, яка реалізована на двох елементах І. Результати попереднього порівняння  $y_{1(i-1)}$  і  $y_{2(i-1)}$  зберігаються в тригерах, запис в які блокується лог.0 з інверсних виходів цих тригерів. Перед початком сортування нового масиву даних тригери результатів порівняння встановлюються в нуль. В інших типах ПЕ числа надходять і порівнюються групами, розрядність яких є  $1 < p < n$ .

Другим варіантом узгодження інтенсивності надходження даних із інтенсивністю сортування є використання гібридних алгоритмів, які ґрунтуються на об'єднанні методів сортування злиттям та підрахунку. При реалізації такого гібридного алгоритму сортування даних здійснюється за методом злиття на ПЕ, які виконують сортування групи чисел методом підрахунку.

Повільнішими є пристрої, які апаратно реалізують проекції потокового графа алгоритму паралельного сортування методом злиття і на вертикальну, і на горизонтальну осі. Структура пристрою, що апаратно реалізує проекцію потокового графа алгоритму паралельного сортування методом злиття масиву із 16-ти чисел на горизонтальну вісь  $X$ , наведена на рис. 5, де ПФІFO – пам'ять FIFO з програмованою затримкою, БУ – блок управління.

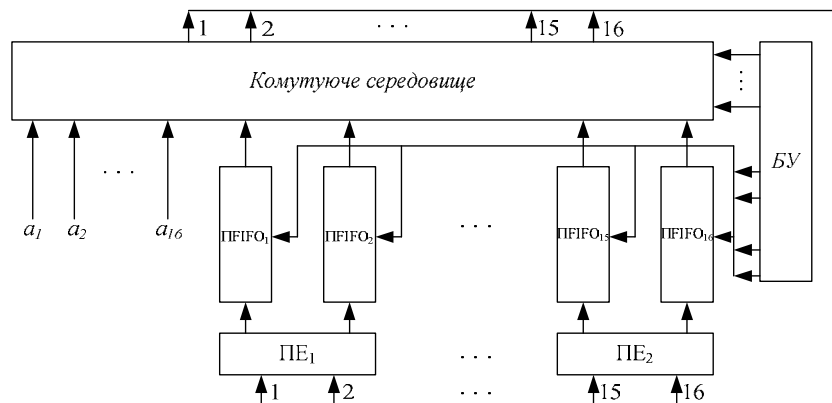


Рис. 5. Структура рекурсивного пристрою сортування масивів даних

Особливістю структури такого пристрою сортування даних є система обернених зв'язків, яка робить його рекурсивним. Основою рекурсивного пристрою сортування даних є масив ПЕ, структура яких наведена на рис. 4, а. Сортування у такому пристрої відбувається по тактах, кількість яких визначається кількістю ярусів потокового графа алгоритму. У кожному  $j$ -му такті роботи рекурсивного пристрою сортування даних виконуються функціональні оператори  $j$ -го ярусу  $\Phi_{ji}$ . Для забезпечення просторо-часового розгортання потокового графа сортування даних використовуються БУ, ПFIFO і комутуюче середовище, які забезпечують відповідно формування сигналів управління, затримку даних на необхідну кількість тактів і комутацію каналів передавання даних у відповідності до потокового графа сортування даних [15]. Сортування масиву із 16-ти чисел у рекурсивному пристрої виконується за 15 тактів. Рекурсивний пристрій сортування даних доцільно використовувати для сортування одиночних масивів даних.

Для сортування інтенсивних потоків даних доцільно використовувати потокові структури, які є апаратним відображенням проекції потокового графа алгоритму паралельного сортування методом злиття на вертикальну вісь  $Y$  (рис.3). Потоківі структури пристроїв сортування даних на основі двошляхового злиття реалізуються на базі ПЕ, які відрізняються тільки обсягом спеціалізованої пам'яті. Структура  $l$ -го, де  $l=1, \dots, k$ , ПЕ одноканального потокового пристрою сортування даних на основі двошляхового злиття наведена на рис.6, де Рг – регістр, Км – комутатор, БУ – блок управління.

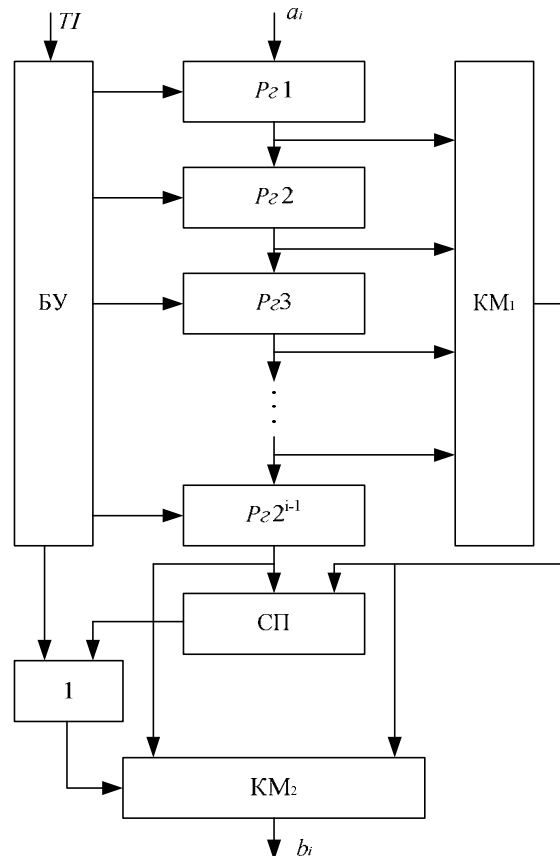


Рис. 6. Структура ПЕ одноканального потокового пристрою сортування даних

Для синтезу одноканального потокового пристрою сортування  $N$  даних необхідно  $k=\log_2 N$  ПЕ [14]. Кожний  $l$ -й ПЕ містить спеціалізовану пам'ять, яка складається з  $2^{l-1}$  регістрів і  $(2^{l-1}-1)$  вхідного комутатора  $Км_1$ . У  $l$ -м ПЕ із двох списків довжиною  $2^{l-1}$  на виході  $Км_2$  формується упорядкований список довжиною  $2^l$ . Управління процесом сортування даних у кожному  $l$ -му ПЕ виконує БУ, який пам'ятає інформацію про всі виконані порівняння та на її основі формує сигнали управління для

Км<sub>1</sub> і Км<sub>2</sub>. Одноканальний потоковий пристрій сортування даних працює за конвеєрним принципом з тактом, що дорівнює  $T_k = t_{P_2} + t_{B_2} + t_{СП} + 2t_{КМ}$ , де  $t_{P_2}$ ,  $t_{B_2}$ ,  $t_{СП}$  і  $t_{КМ}$  – час затримки відповідно регістра, блока управління, схеми порівняння і комутатора. Для сортування масиву даних із  $N$  даних у цьому пристрої необхідно виконати  $\sum_{l=1}^k 2^l$  тактів.

Зменшити час сортування даних можна збільшенням кількості каналів двошляхового злиття. Для реалізації такого підходу необхідно синтезувати багатоканальний потоковий пристрій сортування даних, який апаратно відображає проекцію потокового графа алгоритму паралельного сортування методом злиття на вертикальну вісь  $Y$  (рис. 3, б). Такий пристрій синтезується на базі ПЕ, який складається з таких основних компонентів: спеціалізованої паралельної пам'яті (СПП) і блока паралельного злиття двох груп елементів даних (БПЗ) (рис. 7, а). Необхідна кількість ПЕ для синтезу багатоканального потокового пристрою сортування даних визначається за формулою

$$g = \left\lceil \log_2 \frac{N}{m} \right\rceil.$$

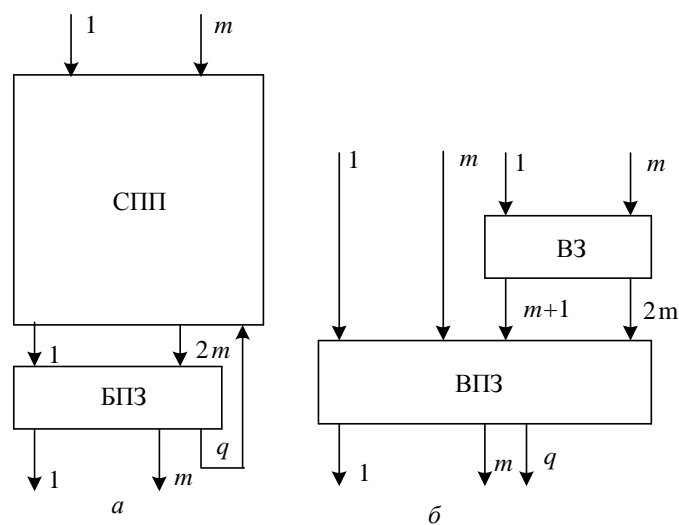


Рис. 7. Структури:  
а – процесорного елемента, б – блока паралельного злиття

Для забезпечення сортування потоків даних у реальному часі, ці компоненти повинні відповідати таким вимогам:

- СПП – здійснювати зберігання, одночасне приймання  $m$  і видавання у  $2m$  елементів даних на вузол паралельного злиття, з яких перші  $m$  – упорядковані елементи масиву  $\{a_i\}_{i=1}^{m2^{p-1}}$ , а другі  $m$  – упорядковані елементи масиву  $\{b_i\}_{i=1}^{m2^{p-1}}$ , де  $p=1, \dots, g$ ;
- БПЗ – виконувати швидке однотактне злиття елементів даних двох упорядкованих груп  $\{a_h\}_{h=1}^m$  і  $\{b_h\}_{h=1}^m$  у масив  $\{c_h\}_{h=1}^{2m}$ , де елемент  $c_m$  супроводжується інформацією про номер входу  $q$ , із якого він надійшов.

Паралельне злиття груп даних  $\{a_h\}_{h=1}^m$  і  $\{b_h\}_{h=1}^m$ , які надходять з виходів СПП, здійснюється блоком паралельного злиття (БПЗ), схема якого наведена на рис.7б, де ВЗ – вузол зсуву, БПЗ – вузол паралельного злиття. Для ефективного паралельного злиття груп даних необхідно попередньо упорядкувати дані, що надходять із  $(m+1)$ -го, ...,  $2m$ -го виходів. Цю функцію для забезпечення високої швидкодії пропонується виконувати на комутаторах. При цьому час упорядкування даних ВЗ буде визначатися часом затримки проходження інформації через комутатор. Для реалізації БПЗ необхідно вибрати швидкодіючий алгоритм паралельного злиття упорядкованих груп даних. Аналіз методів і алгоритмів паралельного сортування даних показав [1,14], що найшвидше апаратна

реалізація одноканального паралельного злиття двох упорядкованих груп даних виконується за методом підрахунку. Час такого злиття визначається за формулою

$$t_{ПЗ} = t_{СП} + t_{СМ} + t_{Дш} + t_{КМ},$$

де  $t_{СП}$  – час парного порівняння;  $t_{СМ}$  – час додавання;  $t_{Дш}$  – час дешифрації;  $t_{КМ}$  – час затримки інформації на комутаторі.

Структура СПП для  $p$ -го ПЕ наведена на рис.8, де  $k=1, \dots, Q/m$ , ЗС – запам'ятовуюче середовище; БП – блок пам'яті; КП – комірка пам'яті; КС – комутуюче середовище; ВК – вузол комутації; Км – комутатор; ВКП – вузол керування пам'яттю.

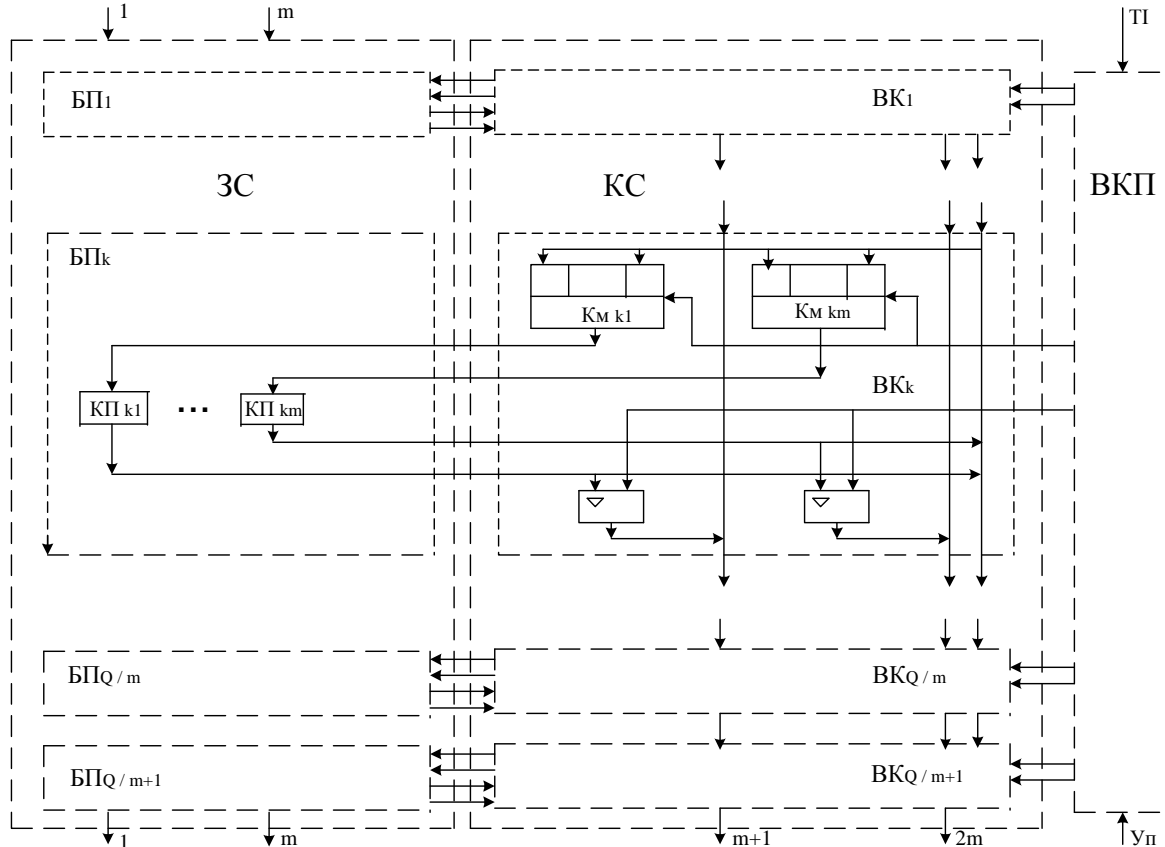


Рис. 8. Схема спеціалізованої паралельної пам'яті

Особливістю СПП є послідовно-паралельна організація. Послідовне з'єднання КП між собою здійснюється за допомогою  $(m+1)$ -входових комутаторів Км, які можуть передавати інформацію без зсуву або із зсувом від 1-го до  $m$  елементів даних [5,16]. Паралельне з'єднання КП стовпців запам'ятовуючого середовища, яке здійснюється за допомогою буферних підсилювачів, забезпечує видавання  $m$  елементів масиву  $\{b_i\}_{i=1}^{m2^{p-1}}$ . Формування сигналів для паралельного читання  $m$  елементів масиву  $\{b_h\}_{h=1}^m$  і слів керування для всіх Км здійснюється ВКП.

СПП працює так. У кожному  $k$ -му такті роботи СПП здійснюється запис  $m$  елементів даних у БП<sub>1</sub> і читання  $2m$  елементів даних, з яких  $m$  перших є елементами масиву  $\{a_i\}_{i=1}^{m2^p}$ , а  $m$  других – елементами масиву  $\{b_i\}_{i=1}^{m2^{p-1}}$ . На основі інформації, яка надходить з входу Уп, у ВКП здійснюється обчислення адреси для  $(k+1)$ -го паралельного читання  $m$  елементів масиву  $\{b_h\}_{h=1}^m$ . Адреса КП з найменшим елементом для  $(k+1)$ -го читання обчислюється за формулою:

$$A_{(k+1)0} = A_{k0} + z_k,$$

де  $A_{k_0}$  – адреса найменшого елемента при  $k$ -му читанні,  $z_k$  – величина зсуву елементів масиву  $\{b_i\}_{i=1}^{m2^p}$ , яка надходить у ВКП з входу Уп. У  $(k+1)$ -му такті роботи на  $(m+1), \dots, 2m$  виходи СПП будуть надходити дані з КП з номерами  $A_{(k+1)_0}, \dots, (A_{(k+1)_0} + m - 1)$ . Крім того, у ВКП формуються дві групи слів для керування комутаторами КС. При цьому комутатори з номерами  $1, \dots, A_{(k+1)_0} + m$  устанолюється на передавання елементів даних із зсувом на  $m$  елементів, а комутатори з номерами  $(A_{(k+1)_0} + m + 1), \dots, Q$  – на передавання у даних із зсувом на  $z$  елементів. З приходом на вхід ТІ тактового імпульсу у всі КП проводиться запис інформації з виходів відповідних комутаторів і читання даних на виходи  $1, \dots, 2m$ . Необхідно зазначити, що на 1-й, ...,  $m$ -й виходи СПП зчитується  $m$  упорядкованих елементів даних масиву  $\{a_i\}_{i=1}^{m2^{p-1}}$ , найбільший з яких надходить на 1-й вихід, наступний на 2-й вихід і так далі, а на  $(m+1)$ -й, ...,  $2m$ -й виходи –  $m$  елементів даних масиву  $\{b_i\}_{i=1}^{m2^{j-1}}$ , які упорядковані по зменшенню з циклічним зсувом ліворуч на  $L = (A_{(k+1)_0} + m - 1)_{\text{mod } m}$  слів. Для кожного  $p$ -о ПЕ обсяг СПП визначається за формулою  $Q = (2^{p-1} + 1)m$ .

### Висновки та перспективи подальших наукових розвідок

1. Розроблення високоефективних паралельних структур для сортування інтенсивних потоків даних у реальному часі методом злиття найдоцільніше здійснювати при інтегрованому підході, який охоплює методи, алгоритми, структури і НВІС-технологію та враховує особливості конкретного застосування.
2. У матричному пристрої сортування даних узгодження інтенсивності надходження даних із інтенсивністю сортування досягається зміною розрядності каналів надходження даних, розрядності ПЕ та використанням гібридних алгоритмів, які ґрунтуються на об'єднанні методів сортування злиттям та підрахунку.
3. У потоковому пристрої сортування даних узгодження інтенсивності надходження даних із інтенсивністю сортування досягається зміною кількості каналів двошляхового злиття.
4. Розроблено нові алгоритми і структури пристроїв для паралельного та паралельно-потокowego сортування інтенсивних потоків даних методом злиття у реальному часі, в яких за рахунок використання гібридних алгоритмів, зміни кількості каналів і розрядності надходження даних досягається узгодження інтенсивності надходження даних із сортувальною здатністю, що забезпечує підвищення ефективності використання обладнання.

1. Д. Кнут. Искусство программирования, том 3: Сортировка и поиск. – 2-е изд. – М., 2000. – 832 с. 2. Высокопроизводительные вычисления для мнгопроцессорных многоядерных систем // Издательство Московского университета, 2010. – 544 с. 3. Паралельные вычисления на GPU, Архитектура и программная модель CUDA // Издательство Московского университета, 2012. – 336 с. 4. Грушицкий Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608 с. 5. Цмоць І. Г. Інформаційні технології та спеціалізовані засоби обробки сигналів і зображень у реальному часі. – Львів: УАД, 2005. – 227 с. 6. Цмоць І. Г., Антонів В. Я., Парубчак В. О. Паралельно-вертикальне сортування одновимірних даних методом злиття з використанням підрахунку // Збірник наукових праць. Інститут проблем моделювання в енергетиці. Вип. 68, 2013. – С. 92–100. 7. Кун С. Матричные процессоры на СБИС. – М.: Мир, 1991. – 672 с. 8. Цмоць І. Г., Рахман М. Л. Алгоритми та пристрої паралельно-потокowego сортування чисел // Збірник наукових праць Інституту проблем моделювання в енергетиці. Вип. 21. – К., 2001. – С. 183–191. 9. Цмоць І. Г., Рахман М. Л. Паралельні алгоритми та пристрої сортування чисел // Збірник наук. праць ІПМЕ НАН України, Вип. 11. – К., 2001. – С. 83–91. 10. Цмоць І. Г., Батюк А. Є. Алгоритми і конвеєрні пристрої сортування даних в

реальному масштабі часу // Вісник Держ. ун-ту "Львівська політехніка": "Інформаційні системи та мережі", № 330. – Львів, 1998. – С. 247–253. 11. Цмоць І. Г., Пасека Є. М., Зербіно Д. Д. Систолічний пристрій з вертикальним сортуванням потоків даних // Вісник Нац. ун-ту "Львівська політехніка" "Комп'ютерні науки та інформаційні технології" № 598. – Львів, 2007. – С. 31–36. 12. Патент України на винахід № 29700. Пристрій для сортування чисел. Батюк А. Є., Рашкевич Ю. М., Цмоць І. Г. – 2000, Бюл. № 6–11. 13. А. С. 1298737 (СССР). Устройство для сортировки чисел. А. А. Мельник, И. Г. Цмоць / Бюл. изобретений 1987, № 11. 14. Компьютеры на СБИС: В 2-х кн. Кн. 2. Мотоока Т., Хорикоси Х. И др. – М.: Мир, 1988. – 336 с. 15. Цмоць І., Демида Б. Структури пам'яті з дисципліною доступу FIFO // Вісник Держ. ун-ту "Львівська політехніка": "Комп'ютерна інженерія та інформаційні технології" № 386. – Львів, 1999. – С. 21–26. 16. Цмоць І. Г., Демида Б. А. Структури спеціалізованої паралельної пам'яті високопродуктивних процесорів управління та цифрової обробки сигналів // Вісник Держ. ун-ту "Львівська політехніка"; "Комп'ютерна інженерія та інформаційні технології", № 380. – Львів, 1999. – С. 18–29. 17. Вовк О. О., Астраханцев А. А. Розроблення методики оцінювання важливості характеристик стеганографічних алгоритмів // Вісник Нац. ун-ту "Львівська політехніка". – 2014. – № 805. 18. Адаптивний синтез формул абстрактних алгоритмів / А. С. Василюк, Т. М. Басюк // Вісник Нац. ун-ту "Львівська політехніка". – 2012. – № 743: Інформаційні системи та мережі. – С. 35–43.