

Корисна модель відноситься до систем пам'яті обчислювальних машин, пристроїв оперативної пам'яті систем та комплексів, і може бути використана при побудові високоефективних систем пам'яті з реалізацією функцій зберігання та обробки інформації усередині пам'яті в широкому діапазоні структур обчислювальних засобів.

Відомо, що час, необхідний для передачі даних з пам'яті в процесор, а результатів обробки - назад в пам'ять, звичайно, буває в кілька разів більше, ніж "чистий" час обробки даних у процесорі, що істотно знижує продуктивність системи в цілому. Крім того, вся історія розвитку інтегральної технології показала, що швидкість роботи (тактова частота) процесора збільшується в два рази приблизно кожні вісімнадцять місяців, а швидкість роботи оперативної пам'яті подвоюється приблизно лише кожні десять років. Це, природно, призводить до збільшення розбіжності між швидкістю процесора та пам'яті, і, зрештою, приводить до так названого "границі" пам'яті, коли процесор буде невиправдано довго простоювати через недостатню швидкість доступу до пам'яті, незважаючи на використання різних хитрувань, наприклад, багаторівневої пам'яті типу кеш та ін. Тому розробка оптимального інтерфейсу процесор-пам'ять була й залишається однією з головних проблем для EOM і комп'ютерних систем різних типів.

Інтерфейс процесор-пам'ять характеризується двома параметрами: часом очікування та шириною смуги пропущення. Час очікування - час між ініціюванням запиту пам'яті процесором і його завершенням. Ширина смуги пропущення - це швидкість передачі, з якої інформація може бути передана (переміщена) до системи пам'яті або від системи пам'яті.

Застосування ієрархії пам'яті забезпечує зменшення середнього часу очікування та знижує вимоги до ширини смуги пропущення. Тому основні зусилля розроблювачів були спрямовані на ефективне використання ієрархії пам'яті.

У той же час було відзначено, що чипи пам'яті, якщо не виходити за їхні межі, мають величезну внутрішню ширину смуги пропущення. Тому багато дослідників провели експерименти з напівпровідниковими приладами, об'єднавши як логічні блоки, так і статичні комірки оперативного запам'ятовувального пристрою (ОЗП) на одному чипі. У цей час, практично всі сучасні мікропроцесори містять високошвидкісну статичну кеш-пам'ять типу SRAM на чипі.

Але тільки недавно успіхи мікроелектроніки уможливили інтеграцію на одному кристалі логічних схем із блоками комірок динамічної пам'яті типу DRAM відносно великої ємності, забезпечивши при цьому основу для практичної реалізації архітектури типу "Процесорів-у-пам'яті" ("Processor-In-Memory" - PIM). Ця архітектура є однією з найефективніших у цей час архітектур комп'ютерних систем, тому що вона встановлює баланс між часом обробки даних і часом доступу до пам'яті за цими даними.

Один з перших побудованих пристроїв пам'яті був C-RAM (Elliott D.: Computational Ram: A Memory-SIMD Hybrid and its Application to DSP. In Proceedings of the Custom Integrated Circuits Conference, Boston, MA, 3 (1992)), розроблений в університеті Торонто в 1992. Подібний проект - Terasys PIM пристрій (Gokhale M., et al.: Processing in Memory: The Terasys Massively Parallel PIM Array. IEEE Computer (April 1995) 23-31) був анонсований в 1995р. Далі з'явилася безліч проектів, які стали досліджувати цю технологію для створення систем різного призначення (наприклад, Processor-In-Memory (PIM) Chips and the TERASYS Workstation.- <http://www.nsa.gov/programs/tech/factsheets/pimchips.html>; Roth Luke, Coraor Lee, Landis David, Hulina Paul, and Deno Scott. Computing in Memory Architectures for Digital Image Processing.- <http://www.csd.cmu.edu/computer.org/comp/proceedings/mtdt/1999/0259/00/02590008abs.htm>; Sterling Thomas L. and Zima Hans P. Gilgamesh: A Multithreaded Processor-In-Memory Architecture for Petaflops Computing. - <http://www.sc-2002.org/paper/dfs/pap.pap.105.pdf>).

Інтеграція множини процесорів на одному чипі з пам'яттю досить великого обсягу пропонує кілька потенційних переваг для комп'ютерних систем у порівнянні з розміщенням процесора та пам'яті на окремих кристалах. До таких переваг насамперед варто віднести:

По-перше, забезпечується широка смуга пропущення між центральним процесором і пам'яттю, оскільки пам'ять типу DRAM розташована на тім же самому чипі, що й обчислювальна логіка (процесор). Архітектура PIM-систем використовує можливості широкої смуги пропущення пам'яті за рахунок звернення до повного рядка пам'яті одночасно та поділу загальної пам'яті чипа на множину незалежно доступних блоків. Через паралелізм доступу до даних усім рядком (наприклад, 2048 біт на звичайному чипі динамічної оперативної пам'яті) і множини банків пам'яті на одному чипі, пікова смуга пропущення пам'яті на чипі може перевищити на два порядки величини смуги пропущення звичайних систем пам'яті аналогічної ємності. Щоб використати доступну смугу пропущення пам'яті, широкий арифметико-логічний пристрій повинен бути інтегровано з буфером рядка кожного банку пам'яті для обробки даних усього рядка.

По-друге, фізичний розмір, вага й відповідно споживана потужність системи в цілому можуть бути істотно зменшені, оскільки більше функцій об'єднано на кожному чипі, і тому менше чипів необхідно для реалізації всього проекту. Щільність виготовлення тепер стає досить великою, щоб розмістити різні функціональні набори в просторі пам'яті одиночної DRAM. Крім того, нові можливості нових кремнієвих технологій виготовлення забезпечують необхідні умови для поліпшення експлуатаційних показників проектів "система на кремнії". З'являється можливість інтегрувати будь-яку комбінацію процесорних ядер (спрощеного варіанта процесора), замовленої логіки, FPGA комірок, і аналогову сукупність схем - всі необхідні вузли (компоненти) для побудови обчислювальних систем на одній підкладці великої інтегральної схеми (BIC). Ці вузли зв'язані дуже швидким широкосмужним інтерфейсом і можуть бути запрограмовані з використанням стандартної поділюваної пам'яті або паралельних алгоритмів передачі повідомлень.

У - третіх, щільне розміщення на чипі широкого арифметико-логічного пристрою з буфером рядка пам'яті дозволяє одержати малий час очікування доступу до даних. Оскільки багато операцій із застосуванням пам'яті можуть бути виконані на PIM - чипі безпосередньо, при цьому кількість переміщень даних за межі кристала може бути істотно зменшено, що призводить відповідно до істотного зменшення споживаної потужності й скороченню конфліктів для віддаленого доступу.

У - четвертих, з'являється можливість використання віртуальності PIM (у плані фізичної переадресації) і обробки довгого слова, спрощується ведучий процесор, спрощується структура пам'яті типу кеш, що також призводить до зменшення споживаної потужності (іноді майже на порядок) і до зниження вартості виробів у порівнянні з виробами, у яких застосовуються роздільні BIC пам'яті та BIC процесора.

Зазначені вище й деякі інші переваги архітектур типу PIM у порівнянні із класичними архітектурами дозволяють стверджувати, що проблема побудови комп'ютерних систем типу "Процесор-у-пам'яті" є досить актуальною.

Відома масштабована архітектура з мультипроцесорною обробкою на одному чипі (Barroso: Luiz Andre; Gharachorloo; Kourosh; Nowatzky; Andreas. Scalable architecture based on single-chip multiprocessing. United States Patent. №6,668,308. Intern'l Class: G06F012/00, U.S. Class: 711/141, December 23, 2003), що включає множину процесорних ядер, підсистему міжз'єднань і комутатор усередині чипа, один або більше контролерів пам'яті, перший і другий рівні кеш-пам'яті, причому кеш-пам'яті першого рівня містять кеш-пам'ять команд і кеш-пам'ять даних для кожного процесорного ядра, а другий рівень кеш-пам'яті логічно розділений серед множини процесорних ядер, що утворюють множину розділених модулів, кожна множина таких модулів має спеціалізований інтерфейс до одного або більшої кількості контролерів пам'яті, які забезпечують оперативну комунікацію дворівневої ієрархії кеш-пам'яті з пам'яттю поза кристалом.

Ця система орієнтована на комерційні робочі навантаження типу мережної діалогової обробки запитів (OLTP), які мають специфічні особливості, що призводять до неефективного використання пам'яті великої ємності, високого рівня невключення (пропусків) при зверненні за даними в кеш-пам'ять, невисокого паралелізму рівня команди та ін. Як приклад реалізації багатопроцесорної системи на одному чипі розглядається проблемно-орієнтована система типу PIRANHA.TM, архітектура якої включає вузол обробки, що містить вісім специфічних процесорних ядер типу ALPHA.TM, окремі кеш-пам'яті команд і кеш-пам'яті даних для кожного ядра, кешей другого рівня, вісім контролерів пам'яті із двома механізмами протоколу когерентності та мережний маршрутизатор.

Однак, незважаючи на можливість масштабування системи й застосування промислового стандарту ASIC - логічних схем, що перебудовуються, типу ПЛІС, які застосовуються для скорочення строків розробки, запропонована архітектура орієнтована на конкретний клас завдань і не може бути перебудована на ефективне використання при рішенні інших класів завдань (наприклад, завдань обробки тривимірної графіки, радарних сигналів та ін.). Отже, архітектурно-структурні рішення не є типовими (універсальними), що призводить до невисокого ринкового попиту на такий тип чипа й, отже, до малої серійності його промислового виробництва та підвищеної вартості.

Крім того, відповідно до запропонованої в даному аналогу структурної схеми процесорного чипа, основна DRAM-пам'ять системи перебуває за межами чипа. Це в порівнянні із системою, у якій така пам'ять розміщена усередині процесорного чипа, призводить до більш високої споживаної потужності за рахунок необхідності збільшення потужності сигналів на вихідних і вхідних контактах чипа, а також до зменшення смуги пропускання при передачі сигналів по ланці основна пам'ять - процесор, і, отже - до зниження швидкодії системи в цілому.

Відома архітектура на чипі динамічної оперативної пам'яті (Saulsbury; Ashley (Los Gatos, CA); Nettleton; Nyles (Campbell, CA); Parkin; Michael (Palo Alto, CA). VLIW computer processing architecture with on-chip dynamic RAM. United States Patent №6,631,439. Intern'l Class: G06F012/00, U.S. Class: 711/104, October 7, 2003), що містить ядро обробки, яке включає один або більше конвеєрів обробки, не менш чим один банк динамічної пам'яті типу DRAM, інтерфейс вводу - виводу для зв'язку з іншими подібними процесорними чипами або сумісними пристроями вводу - виводу, перший контролер пам'яті, з'єднаний з ядром обробки й банками пам'яті, другий контролер пам'яті з'єднаний з першим контролером пам'яті й інтерфейсом вводу - виводу, інтерфейс зовнішньої пам'яті, з'єднаний з першим контролером пам'яті, завантажувальний інтерфейс, з'єднаний із процесорним ядром для завантаження команд у кеші процесорного ядра, діагностичний інтерфейс, з'єднаний із процесорним ядром для забезпечення можливості діагностики внутрішнього стану ядра обробки. Команда обробки містить N команд із P-bit, з'єднаних у лінію разом, що створюють командне слово (VLIW), а процесорне ядро включає один або більше конвеєрів обробки, що мають N процесорних частин, що обробляють N команд із P-bit паралельно. При цьому конвеєр обробки містить виконавчий модуль, що включає цілочисельний процесор, процесор завантаження/зберігання, процесор плаваючої коми, або будь-яку комбінацію одного або декількох цілочисельних процесорів, процесорних пристроїв завантаження й процесорів плаваючої коми. Однак така комбінація задається при виготовленні чипа й не може бути змінена надалі при його експлуатації, що також як і для першого аналога, призводить до невисокого попиту на такі вироби й, отже, до високої його вартості. При цьому кількість конвеєрів обробки для виготовленого пристрою визначається необхідною продуктивністю при рішенні конкретного класу завдань і можливостями мікроелектроніки. Таким чином, ефективність використання обробної системи визначається орієнтацією чипа при його виготовленні на клас розв'язуваних завдань, а також паралельним підключенням таких же чипів, що не завжди можливо й доцільно.

Крім того, у відповідності зі специфічними особливостями архітектурно-структурної організації даного аналога, підкоманди в слові команди VLIW запускаються одночасно, але обробка кожної підкоманди може завершуватися в різний час або в різні тактові цикли, тому що різні типи команд можуть мати різну затримку обробки. При цьому якщо один зі шляхів обробки зупинений, виконання всіх підкоманд буде зупинено до з'ясування стану всіх шляхів обробки. Це призводить до зниження продуктивності системи й до ускладнення процедури компіляції для різних виконуваних застосунків.

Відома система пам'яті з реалізацією функцій зберігання й обробки інформації на одному кристалі, що обрано нами як прототип (Iobst: Kenneth W.; Resnick; David R; Wallgren; Kenneth R. Reconfigurable memory processor. United States Patent №5,396,641, Intern'l Class: G06F013/00, U.S. Class: 713/100, March 7, 1995). Прототип містить: масив пам'яті ємністю (M×N) біт, де M - кількість рядків, N - кількість стовпців, дешифратор рядків, N однорозрядних процесорів по кількості стовпців масиву пам'яті, схему виявлення й виправлення помилок, регістр рядка розрядністю N, селектор читання для виділення читаного слова даних розрядністю r ( $r \leq N$ ) з N-розрядного регістра рядка, дешифратор запису для розміщення при записі r - розрядного слова даних на відповідну позицію N-розрядного регістра рядка, "глобальну" мережу для зв'язку процесорів між собою за принципом "багато - до - одного" або "один - до - багатьох" й "префіксна" мережа для зв'язку процесорів за принципом "багато - до - багатьох". При цьому вихід кожного процесора з'єднаний з виходом однойменного розряду регістра рядка й підключений до першого входу відповідного розряду схеми виявлення й виправлення помилок, однойменний вихід якого підключений до входу відповідного розрядного стовпця масиву пам'яті. Перший вихід кожного стовпця масиву пам'яті з'єднаний із другим входом однойменного розряду схеми виявлення й виправлення помилок,

другий однойменний розрядний вихід якого підключений до першого входу відповідного розряду регістра рядка, другий вихід якого з'єднаний з відповідним першим входом селектора читання й підключений до першого входу відповідного процесора, при цьому другі та наступні виходи всіх процесорів, а також другі та наступні їхні входи з'єднані відповідними зв'язками за допомогою "глобальної" й "префіксної" мереж комутації, на відповідні входи яких підключені ланцюги керування, які підключені до входів "Команда" чипа системи пам'яті. Група других входів "Вибір рядка" масиву пам'яті підключена до відповідних виходів дешифратора рядка, на перші входи якого підключені входи "Адреса рядка" чипа системи пам'яті. Другі входи регістра рядка з'єднані з першими виходами дешифратора запису, перші входи якого підключені до входів "Дані на запис" чипа системи пам'яті, а другі його входи - з'єднані із входами "Адреса стовпця" чипа системи пам'яті та із другими входами селектора читання, перші входи якого підключені до виходів "Читання даних" чипа системи пам'яті, входи "Режим роботи" якого з'єднані з відповідними входами пристрою керування системою пам'яті.

Множина названих пристроїв пам'яті й процесорів об'єднані в групи, кожна група, позначена як процесор що реконфігурується, містить  $\mu$  пристроїв пам'яті,  $\mu$  процесорів,  $\mu$  мультиплексорів, один селектор й один дешифратор мультиплексорів, з'єднані між собою та з іншими вузлами системи відповідними зв'язками. При цьому виходи кожного пристрою пам'яті з'єднані з відповідними входами селектора й з першими входами мультиплексора, виходи якого підключені до входів цього ж пристрою пам'яті, а другі входи мультиплексора підключені до виходів відповідного процесора, входи кожного процесора підключені до відповідних виходів селектора, другі входи якого з'єднані із входами "Вибір режиму" даної групи. Треті входи всіх мультиплексорів підключені до відповідних виходів дешифратора мультиплексорів, входи якого з'єднані із входами "Зовнішнє керування" даної групи.

Прототип має наступні недоліки.

Кожен процесор у системі-прототипі являє собою однорозрядний послідовний пристрій, при цьому всі процесори ідентичні й виконують ту ж саму операцію над різними даними, тобто реалізована архітектура типу ОКМД (одна команда - множина даних). Тому всі архітектурно-структурні рішення й методи організації обчислювального процесу орієнтовані на цей тип архітектури, що відповідає досить вузькій області застосування, призводять до невеликого ринкового попиту на даний тип чипа й, отже, до невеликої кількості виробництва та до високої продажної ціни.

Організація обчислювального процесу виконана в прототипі таким чином, що при записі інформації в пам'ять й читання інформації з пам'яті завжди відбувається зчитування повного N-розрядного слова, хоча для виконання конкретної арифметичної або логічної операції часто потрібні r-розрядні слова ( $r < N$ ). При цьому команду виконують всі процесори над всіма розрядами довгого N-розрядного рядка, але більшість процесорів (крім задіяних з потрібними даними) не зберігають результати обчислень, тобто більша частина процесорів працюють вхолосту, записуючи зворотно в пам'ять тільки що зчитану з її інформацію. Крім того, передача інформації по ланцюгам глобальної або префіксної мережі (від процесора ліворуч до процесора праворуч і навпаки) здійснюється в прототипі за допомогою цих же процесорів, які витрачають істотну частку часу на організацію пересилань, замість виконання арифметичних і логічних операцій, необхідних для рішення конкретного завдання. Усе це призводить до невисокого коефіцієнта завантаження процесорів корисною обчислювальною роботою, до складних проблем по розпаралелюванню процесів і до невисокої продуктивності системи в цілому при рішенні користувальницьких завдань. Крім того, марна робота пам'яті по зчитуванню й запису незадіяних надалі розрядів N-розрядного слова та холоста робота процесорів по виконанню команд над цими розрядами призводить до не виправдано високої споживаної потужності й до підвищеного рівня перешкод усередині чипа й, отже, до зниження надійності системи.

В основу корисної моделі поставлено завдання створити таку систему пам'яті, що завдяки введенню нових вузлів і блоків і зв'язків між ними забезпечує:

- розширення функціональних можливостей системи для збільшення масовості виробництва й зниження вартості кристала системи пам'яті шляхом введення в систему пам'яті набору проблемно-орієнтованих на широкий клас завдань ведучих процесорів;
- підвищення продуктивності системи шляхом введення в систему засобів вибору оптимальних по параметрах ведучих процесорів для кожного застосування й необхідних для них ресурсів, а також шляхом збільшення коефіцієнта завантаження кожного процесора корисною (обчислювальною) роботою й появи внаслідок цього можливостей більш глибокого розпаралелювання процесів;
- зменшення споживаної потужності й рівня перешкод усередині кристала за рахунок виключення марної роботи процесорних ядер і вузлів пам'яті шляхом зчитування й обробки тільки потрібних k-розрядних слів замість істотно більшого N-розрядного рядка.

Поставлене завдання вирішується системою пам'яті на одному кристалі, що містить процесорні ядра, контролер пам'яті, паралельний інтерфейс до зовнішньої пам'яті, вузол інтерфейсів й обробки керуючого пакета, селектор читання, декодер запису, інтерфейс вводу-виводу, інтерфейс завантаження, інтерфейс діагностики та налагодження, регістр рядка, схему виявлення й виправлення помилок і масив оперативної пам'яті, яка організована таким чином, що масив пам'яті ємністю ( $M \times N$ ) біт, де M - кількість рядків, N - кількість стовпців (розрядів), що містить m банків пам'яті розрядністю R ( $R = N/m$ ), групується разом із процесорними ядрами в підбанки пам'яті ємністю ( $M \times r$ ) біт з кількістю стовпців (розрядів) r, так що кожен банк пам'яті містить q підбанків пам'яті ( $q = R/r$ , де q - ціле число,  $R \geq r \geq 1$ ) і q процесорних ядер по одному процесорному ядру для кожного підбанку пам'яті. При цьому в систему пам'яті введені провідні процесори кожний з кеш-пам'яттю команд і кеш-пам'яттю даних, а також перший, другий і третій селектори вибору, котрі з'єднані між собою й з іншими вузлами системи пам'яті відповідними зв'язками. У кожен підбанк пам'яті, що містить масив підбанку пам'яті з відповідною буферною r-розрядною схемою та дешифратором рядків на M виходів, процесорне ядро з кеш-пам'яттю команд, уведений r-розрядний регістр із вхідною логікою та вихідною логікою, з'єднаний із зазначеними й іншими вузлами підбанку пам'яті відповідними зв'язками. У кожен банк пам'яті, уведений один комутатор, що має q входів розрядністю r та q виходів розрядністю r, з'єднаний по входах і виходам з уведеними регістрами та процесорними ядрами підбанків банку пам'яті відповідними зв'язками.

На Фіг.1 представлена схема системи пам'яті з інтеграцією функцій зберігання та обробки інформації на одному кристалі.

На Фіг.2 представлена схема R-розрядного банку пам'яті з q підбанками розрядністю g.

На Фіг.3 представлена схема першого селектора вибору для провідних процесорів.

На Фіг.4 представлена схема другого селектора вибору для банків пам'яті.

На Фіг.5 представлений приклад карти розподілу ресурсів системи пам'яті для чотирьох процесорів і чотирьох банків пам'яті в паралельних гілках А, В, С, D реалізації алгоритму.

На Фіг.6 представлений приклад можливої реалізації ланцюжків макрооперацій на шкалі часу для одного банку пам'яті.

На Фіг.7 представлений приклад можливого суміщення першого та п'ятого, а також третього та шостого ланцюжків макрооперацій згідно Фіг.6 на тому самому обладнанні банку пам'яті для однієї паралельної гілки реалізації застосування.

Система пам'яті з інтеграцією функцій зберігання й обробки інформації на одному кристалі, містить (Фіг.1) масив оперативної пам'яті ємністю  $(M \times N)$  біт, де M - кількість рядків, N - кількість стовпців (розрядів), що містить m банків пам'яті даних 1 ємністю  $(M \times R)$  біт розрядністю R ( $R=N/m$ ) і один банк пам'яті контрольних розрядів 2, контролер пам'яті 3, паралельний інтерфейс до зовнішньої пам'яті 4, вузол інтерфейсів й обробки керуючого пакета 5, селектор читання 6, декодер запису 7, інтерфейс вводу-виводу 8, інтерфейс завантаження 9, інтерфейс діагностики й налагодження 10, регістр рядка 11, схему виявлення та виправлення помилок 12, ведучі процесори 13 кожний з кеш-пам'яттю команд 14 і кеш-пам'яттю даних 15, що входять у вузол кеш-пам'яті 16, перший селектор вибору 17, другий селектор вибору 18, третій селектор вибору 19, а також наступні входи та виходи кристала (чипа) системи пам'яті: входи "Керуючий пакет", входи "Адреса рядка", входи "Адреса стовпця", входи "Завантаження", входи-виходи "Ввід-вивід", входи-виходи "Зовнішня пам'ять", входи-виходи "Діагностика-налагодження".

Перший вихід і перший вхід паралельного інтерфейсу зовнішньої пам'яті 4 з'єднані відповідно з першим входом і першим виходом контролера пам'яті 3, другий вихід і другий вхід якого підключений відповідно до першого входу й першого виходу інтерфейсу вводу-виводу 8, другі виходи якого з'єднані із другими виходами паралельного інтерфейсу до зовнішньої пам'яті 4 і підключені до перших входів декодера запису 7, перші виходи якого з'єднані з першими входами регістра рядка 11, перші виходи та другі входи якого підключені відповідно до перших входів і перших виходів селектора читання 6. Другі виходи селектора читання 6 з'єднані із другими входами інтерфейсу вводу-виводу 8 і із другими входами паралельного інтерфейсу зовнішньої пам'яті 4, треті входи й треті виходи якого підключені до входів-виходів "Зовнішня пам'ять" чипа системи пам'яті, а входи-виходи "Ввід-вивід" чипа системи пам'яті з'єднані із третіми входами й третіми виходами інтерфейсу вводу-виводу 8. Перші виходи й перші входи схеми виявлення й виправлення помилок 12 підключені відповідно до третіх входів і других виходів регістра рядка 11, треті виходи якого з'єднані із другими входами селектора читання 6. Треті входи селектора читання з'єднані із третіми входами декодера запису 7 і підключені до входів "Адреса стовпця" чипа системи пам'яті. Входи "Діагностика/Налагодження" чипа системи пам'яті підключені до перших входів і до перших виходів інтерфейсу діагностики й налагодження 10, а входи "Керуючий пакет" з'єднані з першими входами вузла інтерфейсів й обробки керуючого пакета 5. Входи "Завантаження" чипа системи пам'яті з'єднані з першими входами інтерфейсу завантаження 9, другі входи якого з'єднані із третіми виходами контролера пам'яті 3, а перші виходи інтерфейсу завантаження 9 підключені до перших входів "Завантаження" банків пам'яті 1 й 2.

Перші виходи вузла інтерфейсів й обробки керуючого пакета 5 з'єднані з першими входами першого селектора вибору 17, перші виходи якого підключені до перших входів відповідних ведучих процесорів 13, другі входи й перші виходи кожного ведучого процесора 13 з'єднані відповідно з першими виходами й першими входами вузла кеш-пам'яті 16, другі виходи якого підключені до відповідних четвертих входів декодера запису 7, п'ятий вхід якого з'єднаний з першим виходом третього селектора вибору 19 і з четвертим входом селектора читання 6, треті виходи якого підключені до відповідних других входів вузла кеш-пам'яті 16. Треті входи вузла кеш-пам'яті 16 з'єднані із другими виходами інтерфейсу завантаження 9, треті виходи якого підключені до других входів першого селектора вибору 17, третій вхід якого з'єднаний із другим виходом інтерфейсу діагностики й налагодження 10. Третій вихід інтерфейсу діагностики й налагодження 10 підключено до першого входу другого селектора вибору 18, а четвертий вихід з'єднано з першим входом третього селектора вибору 19, другий вхід якого з'єднано із другим виходом вузла інтерфейсів й обробки керуючого пакета 5. Третій вихід вузла 5 підключено до другого входу другого селектора вибору 18, кожен перший вихід якого з'єднано із другим входом ("Вибір банку") відповідного банку пам'яті, а кожні другі виходи третього селектора вибору 19 з'єднані із третіми входами ("Вибір підбанку") відповідних банків пам'яті. Четвертий вихід ("Читання/Запис") контролера пам'яті 3 підключено до відповідних однойменних четвертих входів банків пам'яті 1 й 2, п'яті входи ("Тип А/Тип В") яких з'єднані з однойменними п'ятими виходами контролера пам'яті 3, треті входи якого ("Запит") підключені до однойменних перших входів відповідних банків пам'яті, другі виходи яких з'єднані із другими входами схеми виявлення й виправлення помилок 12. Входи "Адреса рядка" чипа системи пам'яті підключені до відповідних однойменних шостих входів кожного банку пам'яті, при цьому сьомі входи банків пам'яті 1 з'єднані з відповідними четвертими виходами регістра рядка 11, п'яті входи якого підключені до сьомих входів банку пам'яті контрольних розрядів 2. Восьмі входи ("Комутація") кожного банку пам'яті з'єднані з відповідними четвертими виходами вузла інтерфейсів й обробки керуючого пакета 5, п'яті входи якого підключені до четвертих входів інтерфейсу вводу-виводу 8, шості входи - до четвертих входів інтерфейсу зовнішньої пам'яті 4, сьомі входи - до четвертих входів контролера пам'яті 3. Восьмі входи вузла інтерфейсів й обробки керуючого пакета 5 підключені до других входів інтерфейсу діагностики й налагодження 10, а дев'яті входи - до третіх входів схеми виявлення й виправлення помилок 12.

Кожен банк пам'яті (Фіг.2) містить q підбанків пам'яті 20 ємністю  $(M \times r)$  біт з кількістю стовпців (розрядів), рівним r ( $R \geq r \geq 1$ ,  $q=R/r$ , q - ціле число), один комутатор 21, а також наступні входи й виходи: перші входи банку пам'яті "Завантаження", другі входи "Вибір банку пам'яті", треті входи "Вибір підбанку пам'яті", четверті входи "Читання/Запис", п'яті входи "Тип А/Тип В", шості входи "Адреса рядка", сьомі входи, підключені до відповідних виходів регістра рядка 11, восьмі входи "Комутація", перші входи "Запит", другі входи, підключені до відповідних входів схеми виявлення й виправлення помилок 12.

Кожен підбанк пам'яті 20 (Фіг.2) містить масив пам'яті 22 ємністю  $(M \times r)$  біт, дешифратор r-розрядних рядків на M виходів 23, буферну r - розрядну схему 24, процесорне ядро 25, кеш-пам'ять команд 26, регістр 27 підбанку із

вхідною логікою 28 і вихідною логікою 29. При цьому перші виходи й перші входи масиву пам'яті 22 для кожного одного розряду з  $g$  з'єднані відповідно з першими входами й першими виходами буфера рядка 24, виходи дешифратора 23 підключені до входів відповідних рядків масиву пам'яті 22, а перші входи дешифраторів 23 всіх підбанків банку пам'яті підключені до відповідних шостих входів ("Код адреси рядка") цього банку пам'яті, другі однойменні входи дешифратора рядка 23 кожного підбанку підключені до четвертих входів ("Читання/запис") банку пам'яті. Перші виходи процесорного ядра 25 підключені до відповідних перших входів кеш-пам'яті команд 26, перші виходи якої з'єднані з першими входами процесорного ядра 25, другий вихід якого підключено до відповідного першого виходу "Запит" банку пам'яті. Другі входи кожного процесорного ядра з'єднані з відповідними першими виходами комутатора 21, а перші входи "Завантаження" банку пам'яті з'єднані із другими входами кожної кеш-пам'яті команд 26. Треті входи кожного процесорного ядра 25 підбанку пам'яті підключені до перших входів вхідної логіки 28 регістра підбанку 27, другі входи якої з'єднані із другими виходами буферної схеми 24. Другі входи буферної схеми 24 з'єднані з першими виходами вихідної логіки цього регістра, другі входи якої підключені до відповідних перших входів комутатора, при цьому треті (керуючі) входи вхідної логіки 28 і треті (керуючі) входи вихідної логіки 29 всіх регістрів 27 банку пам'яті підключені до відповідних других входів комутатора 21 і до п'ятих входів ("Тип А") банку пам'яті. Четверті (керуючі) входи вхідної логіки 28 і четверті (керуючі) входи вихідної логіки 29 всіх регістрів банку пам'яті також підключені до відповідних п'ятих входів ("Тип В") банку пам'яті. П'яті входи вхідної логіки 28 регістра 27 кожного підбанку з'єднані з відповідними третіми входами процесорного ядра 25 цього підбанку та підключені на відповідні розрядні позиції сьомих входів, які підключені до відповідних виходів регістра рядка 11 (Фіг.2, Фіг.1). Другі входи банку пам'яті з'єднані з відповідними третіми виходами вихідної логіки 29 всіх регістрів 27 банку пам'яті. Кожен третій вхід ("Вибір підбанку") банку пам'яті підключено до п'ятого входу відповідного регістра 27 підбанку та до четвертого входу процесорного ядра 25 цього підбанку, а другі входи ("Вибір банку") банку пам'яті з'єднані із третіми входами дешифраторів 23 рядків підбанку даного банку пам'яті, восьмий вхід ("Комутація") якого підключено до третього входу комутатора 21.

Кожний (перший, другий і третій) селектор вибору (Фіг.3, Фіг.4) містить лічильник фаз 30 гілок алгоритму, що виконуються паралельно, дешифратор констант фаз 31, пам'ять констант фаз 32 і регістр керування вибором 33, при цьому перший вхід лічильника фаз 30 з'єднано з відповідним входом "Ознаки фаз" першої групи входів селектора, підключених до відповідних виходів вузла інтерфейсів та обробки пакета, виходи лічильника фаз 30 підключені до відповідної входу дешифратора констант фаз 31, виходи якого з'єднані з першими входами пам'яті констант фаз 32, другі входи якої підключені до відповідних входів "Завантаження констант" першої групи входів селектора, виходи кожного розряду пам'яті констант фаз 32 з'єднані з першими входами відповідних розрядів регістра керування вибором 33, другі входи кожного розряду цього регістра підключені до відповідних входів другої групи входів селектора, підключених до відповідних виходів інтерфейсу діагностики й налагодження, а виходи кожного розряду регістра керування вибором 33 з'єднані з відповідними виходами групи виходів селектора вибору, які підключені до відповідних входів обраних вузлів (ведучих процесорів або банків пам'яті, або підбанків пам'яті), при цьому треті входи пам'яті констант фаз тільки для селектора вибору провідних процесорів (Фіг.3) підключені до других входів цього селектора, які з'єднані з інтерфейсом завантаження (Фіг.1, Фіг.3).

Розглянемо архітектурно-структурну організацію та принципи функціонування складових вузлів пропонованого корисної моделі.

Засоби зберігання інформації призначені для тимчасового зберігання даних і команд усередині чипа системи пам'яті. Основою середовища зберігання інформації на чипі є масив оперативної пам'яті досить великої ємності  $W$  біт, що виступає в якості основної оперативної пам'яті як для ведучого процесора 13, так і для процесорного ядра 25. Цей масив пам'яті організовано на кристалі у вигляді матриці  $(M \times N) = W$  біт, де  $M$  - кількість рядків, а  $N$  - кількість стовпців (розрядів), що розділена на однакові підматриці першого рангу розмірністю  $(M \times R)$ , де  $R$  - кількість стовпців (розрядів) однієї такої підматриці, що створюють  $D$  банків пам'яті 1 ( $D = N/R$ ) у сукупності з віднесеними до кожної підматриці відповідними вузлами (Фіг.1). При цьому  $R > 1$ .

Крім того, масив основної пам'яті кожного банку 1 також розділений на однакові підматриці 20 (Фіг.2) другого рангу розмірністю  $(M \times r)$ , де  $r$  - кількість стовпців (розрядів) однієї підматриці другого рангу, що створюють  $Q$  підбанків пам'яті 20 у сукупності з віднесеними до кожної такої підматриці відповідними вузлами (Фіг.2). При цьому  $R \geq r \geq 1$ . Наприклад, якщо  $M=1024$ ,  $N=512$ ,  $R=128$ ,  $r=8$ , то пам'ять на чипі ємністю 512 Кбіт містить чотири 128 - розрядних банки пам'яті ємністю 128 Кбіт, при цьому кожен банк пам'яті містить шістнадцять 8-розрядних підбанків пам'яті ємністю 8 Кбіт.

Як масив основної пам'яті, як правило, застосовується динамічна пам'ять (DRAM) досить великої ємності зі швидкістю роботи істотно меншою, чим швидкість роботи логічних схем процесора або статичної (SRAM) - пам'яті. Для узгодження швидкості роботи процесора та DRAM - пам'яті використовується кеш-пам'ять статичного типу, численні рішення по організації якої та методам її застосування широко відомі. У даній корисній моделі, оскільки ставиться мета максимального використання можливих ресурсів для підвищення продуктивності системи, класичний шлях застосування кеш-пам'яті типу SRAM не може бути виключений з розгляду. Тому для узгодження швидкості роботи ведучих процесорів 13 зі швидкістю роботи основного масиву пам'яті застосовано вузол кешей 16, що містить кеш-пам'ять команд 14 і кеш - пам'ять даних 15 (Фіг.1), кожен з яких приєднаний по входах і виходам відповідно до виходів і входів ведучого процесора 13.

Для забезпечення роботи кожного процесорного ядра 25 застосована кеш-пам'ять команд 26. Таким чином, ведучий процесор 13 використовує при роботі дворівневу ієрархічну систему пам'яті, що складається з кеш-пам'яті даних 15 і масивів основної пам'яті 22, а процесорне ядро - масив пам'яті 22. При підключенні до кристала через інтерфейс 4 зовнішньої пам'яті великої ємності виходить, що кожен ведучий процесор працює із трьохрівневою ієрархічною системою пам'яті, а кожне процесорне ядро - із дворівневою.

Архітектурно-структурна організація та режими роботи кеш-пам'яті команд 14, кеш-пам'яті даних 15 вузла 16 (Фіг.1) і кеш-пам'яті команд 26 (Фіг.2) у принципі відомі. Однак, у пропонованій корисній моделі наявність зв'язку з перших виходів інтерфейсу завантаження 9 на перші входи завантаження банків пам'яті (Фіг.1) і далі на другі входи кеш-пам'яті команд 26 (Фіг.2) дозволяє завантажити або перезавантажити програмні модулі завантаження й ініціалізації процесорних ядер 25 як перед початком роботи всієї системи, так і при необхідності в динаміці, коли

згідно алгоритму розв'язуваного завдання необхідно використати додатково наявне процесорне ядро (або паралельно декілька процесорних ядер). Аналогічно, наявність зв'язку із другого виходу інтерфейсу завантаження 9 на третій вхід вузла кешей 16 дозволяє занести в кеш-пам'ять команд 14 програму завантаження та ініціалізації для відповідно обраного ведучого процесора 13.

Крім інформаційних масивів, що перебувають у банках пам'яті 1, є також масив контрольних кодів, які розміщені в банку пам'яті 2 (Фіг.1). Схема організації банку пам'яті 2 ідентична схемі організації банку пам'яті 1. Однак кількість стовпців цього банку й відповідно кількість підбанків визначається кількістю інформаційних розрядів  $N$  повного рядка й прийнятим способом контролю й корекції помилок. З огляду на те, що звичайно рядок містить від декількох сотень до декількох тисяч біт, може виявитися, що навіть при застосуванні одного зі способів контролю - виявлення двох помилок і виправлення однієї, кількість розрядів банку пам'яті 2 буде порівнянн з кількістю розрядів одного банку пам'яті 1.

Тому що будь-який спосіб контролю збільшує час передачі інформації (за рахунок додавання часу, затрачуваного схемою контролю), то для рішення деяких завдань, що вимагають масових пересилань інформації усередині чипа, можна відключати систему контролю, підключаючи її до процесу пересилань тільки при прийомі вхідної інформації ззовні й при видачі її іншим пристроям за межі чипа. Таке рішення дозволяє істотно зменшити (у деяких випадках у кілька разів) час рішення завдання в цілому. У пропонованій корисній моделі цей позитивний ефект досягається шляхом введення наступного нового зв'язку для банку контрольних кодів 2: перші виходи другого селектора 18 з'єднані з відповідними другими входами банку контрольних кодів 2 і далі (за аналогією з банком 1) із третіми входами дешифратора рядка 23, сигнали на вході якого або дозволяють, або блокують роботу банку 2 системи контролю.

Крім того, пропонована корисна модель у порівнянні із прототипом забезпечує можливість налаштування архітектури системи пам'яті шляхом підключення до роботи або відключення від роботи будь-якого банку пам'яті розрядності  $R$ . Тим самим забезпечується можливість гнучкого налаштування розрядності даних і їхньої кількості в межах усього  $N$ -розрядного рядка масиву пам'яті, орієнтуючись на відповідні методи обробки й наявні ресурси для розпаралелювання алгоритму розв'язуваного завдання. При цьому передачі можуть здійснюватися не тільки  $N$ -розрядним рядком, але й довільними кратними  $R$ -розрядними групами (наприклад, частинами рядка розрядністю  $R, 2R, \dots, \alpha R$ ;  $\alpha R \leq N$ ) як зі своїми відповідними контрольними розрядами, так і без них. Для керування такою комутацією використовуються наступні введені зв'язки: треті входи дешифраторів рядка 23 кожного підбанку пам'яті 20 з'єднані й підключені до відповідного другого входу "Вибір банку пам'яті", п'ятий вхід дозволу роботи регістра 27 кожного підбанку пам'яті 20 з'єднано із четвертим входом процесорного ядра 25 цього підбанку й підключено до відповідного входу "Вибір підбанку пам'яті" групи вхідних третіх контактів банку пам'яті.

Регістри 27 із вхідною логікою 28 і вихідною логікою 29, введені в кожен підбанк 1 і підбанк 2 (Фіг.2), утворюють у сукупності  $(N+K)$ -розрядний регістр, що містить не тільки  $N$  інформаційних, але й  $K$  контрольних розрядів. Застосування цих регістрів дозволяє здійснювати комутацію даних між вузлами системи пам'яті як усередині одного банку, використовуючи вхідну 28 і вихідну 29 логіку, так і між банками пам'яті через другі виходи й сьомі входи кожного банку пам'яті, використовуючи регістр рядка 11 (Фіг.1). При цьому передачі можуть здійснюватися не тільки  $R$ -розрядним рядком, але й довільними кратними  $g$  розрядними групами рядка (наприклад, частинами рядка розрядністю  $g, 2g, \dots, \rho g$ ;  $\rho g \leq N$ ) як зі своїми відповідними контрольними розрядами, так і без них. Для керування такою комутацією використовуються наступні введені зв'язки: четвертий вхід вихідної логіки 29 і четвертий вхід вхідної логіки 28 підключено до відповідних входів "Тип В" групи п'ятих входів банку пам'яті; третій вхід вихідної логіки 29 і третій вхід вхідної логіки 28 підключено до відповідних входів "Тип А" групи п'ятих входів банку пам'яті (Фіг.2).

Застосування такої комутації дозволяє більш точно (у порівнянні з налаштуванням, описаним вище), вибирати розрядність даних у межах банку пам'яті, орієнтуючись на відповідні методи обробки й наявні ресурси для розпаралелювання алгоритму розв'язуваного завдання. Наприклад, при рішенні найпростіших завдань обробки чорно-білих зображень може бути достатнім восьмирозрядне слово, кольорових - тринадцятирозрядне слово й т.д. Тим самим з'являється можливість як грубої (дискретністю в один банк пам'яті розрядністю  $R$ ), так і точної (дискретністю в один підбанк пам'яті розрядністю  $g$ ) налаштування архітектури системи пам'яті, вибираючи розрядність і змінюючи кількість паралельних гілок обробки за рахунок одночасного використання більшої кількості даних, розміщених в  $N$ -розрядній зоні масиву пам'яті й у тому числі - у регістрі рядка 11 (Фіг.1, Фіг.2).

Застосування зазначених регістрів дозволяє також збільшити кількість ступенів конвеєрної обробки даних як мінімум на один рівень, що також сприяє підвищенню продуктивності системи в цілому.

Пристрій обробки інформації. У пропонованій корисній моделі застосовані два рівні пристроїв обробки: перший рівень, уведений авторами корисної моделі, - ведучі процесори 13 (Фіг.1), другий рівень - процесорні ядра 25 (Фіг.1, Фіг.2). При цьому кожен ведучий процесор орієнтовано головним чином на реалізацію базових (глобальних) не часто повторюваних фрагментів алгоритму, які є специфічними для конкретного розв'язуваного завдання. При цьому вони працюють із розрядністю слова, що є загальноприйнятою для конкретного типу архітектури процесора й розв'язуваного типу завдання (наприклад, для процесора з плаваючою комою, з фіксованою комою та ін.).

Крім того, ведучий процесор виконує й множину системних функцій по організації й керуванню потоковим процесом обробки й передачі інформації у відповідному банку пам'яті. Зокрема, ведучий процесор аналізує вихідне завдання, розділяючи його на підзавдання для кожного процесорного ядра, що перебуває в банку пам'яті, визначає рівні можливого розпаралелювання й ступінь конвеєризації кожної паралельної гілки, вирішує проблему збалансованого розподілу ресурсів для підключених до нього банків пам'яті, виконує ряд операцій по реструктуризації програм і структуризації результатів реалізованих кожним банком пам'яті застосувань, і інші функції.

Ведучі процесори можуть бути однотипними (з погляду архітектурно-структурної організації) з однаковими параметрами, а також різнотипними, наприклад, проблемно орієнтованими або навіть спеціалізованими. Наприклад, як ведучі процесори можуть виступати процесори з фіксованою комою, процесори із плаваючою комою, спеціалізовані процесори реального часу, процесори обробки зображень, процесори для обробки радарних сигналів і т.д.

Ведучі процесори у зв'язку з покладеними на них функціями є могутнішими (чим процесорні ядра) і тому їхні схеми є більше складними. Тип і кількість провідних процесорів визначається особливостями класу розв'язуваних завдань, а також технологічними можливостями мікроелектроніки, що визначають ступінь інтеграції й можливість розміщення всіх вузлів системи пам'яті на одному чипі.

Процесорні ядра 25 представлені простими схемами обробки інформації (наприклад, у вигляді "усічених" RISC-процесорів) і орієнтовані на реалізацію часто виконуваних (масових) операцій, наприклад, операції порівняння, зсуву, додавання, пошук найбільшого або найменшого значення та ін. Кожне процесорне ядро, що належить підбанку пам'яті 20, має  $g$ -розрядний арифметико-логічний пристрій (АЛП), при цьому  $Q$  процесорних ядер, по одному для кожного підбанку пам'яті, містять відповідно  $Q$  штук АЛП сумарної розрядності  $R$  (Фіг.1), а  $D$  банків всієї системи пам'яті на одному чипі (Фіг.2) містять  $(Q \times D)$  арифметико-логічних пристроїв із загальною розрядністю  $N$  (без урахування контрольних розрядів банку пам'яті 2), тобто рівної розрядності всього рядка інформаційного масиву пам'яті  $M$ . При цьому АЛП кожного процесорного ядра підбанку 20 має доступ до відповідного  $g$ -розрядного регістра 27, а також через схему виявлення помилок 12 - до регістра рядка 11, так що вся сукупність АЛП має доступ до всіх регістрів 27 загальною розрядністю  $(N+K)$  біт, де  $K$  - кількість контрольних розрядів, і відповідно до регістра рядка 11 також розрядністю  $(N+K)$  біт. Тому або всі АЛП, або кілька АЛП, обраних системою комутації (з використанням нових описаних вище зв'язків для вибору банку й підбанку пам'яті) можуть виконувати множину як однакових, так і різних операцій з даними, розміщеними на регістрах 27 або на окремих полях регістра 11 одночасно, що в порівнянні із прототипом підвищує гнучкість системи пам'яті з погляду її проблемної орієнтації на клас розв'язуваних завдань або на конкретне завдання.

Структура АЛП процесорного ядра 25 банку пам'яті може бути виконана в класичному варіанті згідно (Карцев М.А., Брик В.А. Вычислительные системы и синхронная арифметика. -М.: Радио и связь, 1981. -360с.). Однак для пропонованої системи пам'яті може також застосовуватися структура АЛП, орієнтована на одночасну обробку декількох операндів, представлених у вигляді довгого рядка. У цьому випадку АЛП будується у вигляді двох ступеневі конвеєрної схеми по модульному принципові, що здатна виконувати необхідний і достатній набір операцій (додавання/вирахування, множення, операцію згортки, операцію порівняння та ін.) над операндами різної розрядності (наприклад, 8-розрядні, 16 й 32-розрядні). При цьому виконання операцій додавання/вирахування над 8-розрядними, 16-розрядними та 32-розрядними операндами й множення 8-розрядних операндів провадиться на першому щаблі конвеєра, а для виконання більш складних операцій типу множення 16-розрядних й 32-розрядних чисел, операцій згортки й операцій порівняння підключається другий щабель конвеєра, за допомогою якого виробляється формування остаточного результату.

Вузол інтерфейсів та обробки керуючого пакета 32. Тому що архітектурно-структурна організація системи пам'яті в цілому виходить із концепції потокової обробки інформації, то й вузол 5 також заснований на принципах керування поточними процесами обробки. Вихідною інформацією для цього вузла є пакет, що надходить від Host-машини або від інших аналогічних чипів на перший вхід вузла 5 (Фіг.1) і включає значення параметрів і специфікатори дій, що спрямовуються обумовленому адресатові. Значення параметрів можуть використовуватися при виконанні поточної дії, або можуть бути частиною процесу, що ця дія переносить у наступне місце розташування. Специфікатор дії пакета визначає операції, які повинні бути виконані об'єктом-адресатом після одержання пакета. Це можуть бути операції системи команд, наприклад, операції завантаження або додавання, або ідентифіковані методи (процедури).

Пакети також містять додаткові допоміжні поля, необхідні для транспортування, виявлення помилок, маршрутизації й керування контекстом. Пакет може змінюватися по довжині, таким чином, забезпечуючи ефективну обробку простих операцій з маленькими пакетами та ефективним використанням смуги пропускання для переміщення більших блоків даних.

Всі пакети направляються до об'єкта - одержувача або адресатові, всі адреси пакета - віртуальні адреси, які можуть ідентифікувати індивідуальні перемінні, блоки даних, структур, об'єктів або потоків, а також потоків вводу - виводу. Заключний елемент пакета - поле продовження - частина пакета, що визначає, що повинно бути зроблене після закінчення його дії. Іноді дія пакета призводить до створення одного або більше дочірніх пакетів.

Пакети направляють всі транзакції та надають даним і специфікаторам команд відповідні повідомлення, визначаючи послідовність операцій, і рядки даних, над якими повинна бути виконана операція.

Пакети також можуть використовуватися для виконання операцій типу віддаленого завантаження або зберігання, а також для виклику методів обробки на інший аналогічний чип системи пам'яті, переміщаючи тим самим за допомогою пакетів методи обробки ближче до даних за менший час, чим дані (як звичайно) передаються до них. Елементи чипів (наприклад, процесорні ядра), приймаючи передані методи обробки інформації, виконують зазначені дії, що призводять до локальної модифікації їхнього стану, до породження й відправлення нових пакетів до інших чипів. Пакети дозволяють здійснювати розбивку або розщеплення операцій, забезпечуючи тим самим припустимий час очікування для всієї системи. Їхня ефективність істотно підвищується при апаратній підтримці реалізованих ними функцій.

Стосовно до пропонованої системи пам'яті керуючий пакет, крім зазначених вище полів загального призначення, містить поля та ознаки, що відбивають характер і послідовність дій, що забезпечують:

1). Настроювання архітектурно-структурного образу системи перед запуском її на рішення конкретного завдання, а також при необхідності перебудову системи в процесі роботи шляхом підключення відповідного ведучого процесора 13, вибору типу й кількості відповідних банків пам'яті 1, а також, при необхідності, вибору типу й кількості відповідних підбанків пам'яті 20 у складі обраних банків.

2). Різні режими роботи системи пам'яті, у тому числі:

- як звичайна пам'ять із використанням ємності пам'яті, розміщеної тільки на кристалі (чипі);
- як звичайна пам'ять із використанням ємності пам'яті на кристалі й додатковій зовнішній пам'яті, підключеної до чипа через інтерфейс зовнішньої пам'яті 4;

- у якості "Процесора в пам'яті" з використанням ресурсів обробки інформації тільки власного кристала (ведучі процесори 13, процесорні ядра 25 та ін.);

- у якості "Процесора в пам'яті" з використанням власних ресурсів обробки інформації й додаткових ресурсів за рахунок інших чипів, підключених через інтерфейс вводу-виводу 8 та інтерфейс зовнішньої пам'яті 4.

3). Застосування, або відключення банку контрольних кодів 2 (Фіг.1) виявлення та виправлення помилок.

4). Формування керуючих сигналів і сигналів синхронізації для забезпечення роботи всіх вузлів системи пам'яті, у тому числі:

- керування роботою кеш-пам'яті з урахуванням реалізації механізмів когерентності;
- реалізація режимів "Читання/Запис" як при звертанні до пам'яті усередині чипа, так і за його межами через інтерфейс зовнішньої пам'яті 4; реалізація режимів роботи "Тип А", "Тип В" та ін.

Вузол інтерфейсів й обробки керуючого пакета 5 аналізує всі поля пакета, що надходить, та виробляє сигнали синхронізації й керуючі сигнали для першого селектора вибору 17 ведучого процесора 13 (наприклад, ознака закінчення фази роботи ведучого процесора), другого селектора вибору 18 (наприклад, ознака закінчення фази роботи банку пам'яті), третього селектора вибору 19 підбанків пам'яті (наприклад, ознака закінчення фази роботи підбанку пам'яті), сигнали синхронізації й керуючі сигнали для забезпечення працездатності інтерфейсу зовнішньої пам'яті 4, інтерфейсу вводу-виводу 8, контролера пам'яті 3, схеми виявлення та виправлення помилок 12 а також інших схем і вузлів системи пам'яті, зв'язок з якими на Фіг.1 і Фіг.2 не показані.

У цьому випадку під ознакою закінчення фази роботи одного ведучого процесора (верхній рівень - BP на Фіг.3) розуміється зафіксований у часі вузлом 5 (або HOST-машиною) момент закінчення конкретної паралельної гілки (ділянки) алгоритму, що реалізується тільки даним процесором за допомогою виділених для нього ресурсів без залучення інших ведучих процесорів усередині чипа, або додаткових засобів обробки за межами чипа. Моменти закінчення різними ведучими процесорами різних гілок алгоритму (або різних завдань), як правило, не збігаються, тому їхня сукупність за принципом логічного "АБО" утворить послідовність у часі ознак закінчення фаз верхнього рівня ("Ознака фази BP"), що використовується першим селектором 17 (Фіг.1, Фіг.3) для підключення у відповідні моменти часу необхідних ведучих процесорів 13, тобто для настроювання архітектури системи пам'яті в цілому на тип розв'язуваного завдання.

Під ознакою закінчення фази роботи одного банку пам'яті (середній рівень) розуміється зафіксований у часі вузлом 5 (або HOST-машиною) момент закінчення безперервного використання ресурсів даного банку для будь-якого обраного ведучого процесора (або декількох процесорів). Моменти закінчення використання різних банків також можуть не збігатися, тому їхня сукупність за принципом логічного "АБО" утворить послідовність у часі ознак закінчення фаз середнього рівня ("Ознака фази CP"), що використовується другим селектором 18 (Фіг.1, Фіг.3) для вибору необхідних банків пам'яті 1 і банку пам'яті 2, тобто для настроювання архітектури системи пам'яті на особливості конкретного розв'язуваного завдання.

Аналогічно, під ознакою закінчення фази роботи одного підбанку пам'яті 20 (нижній рівень) розуміється зафіксований у часі вузлом 5 (або HOST-машиною) момент закінчення безперервного використання ресурсів даного підбанку для будь-якого обраного банку. Сукупність моментів закінчення використання різних підбанків також утворить за принципом логічного "АБО" послідовність у часі ознак закінчення фаз нижнього рівня ("Ознака фази HP"), що використовується третім селектором 19 (Фіг.1 - Фіг.3) для вибору необхідних підбанків пам'яті 20, тобто для настроювання архітектури системи пам'яті на особливості реалізації конкретних операцій.

Інтерфейс вводу-виводу 8 призначений для зв'язку чипа системи пам'яті з іншими подібними чипами або сумісними з ним пристроями вводу-виводу при наявності керуючого пакета, що надходить на відповідні входи "Керуючий пакет" від HOST-машини або інших подібних чипів. При цьому контролер пам'яті 3 аналізує запити вводу-виводу та повідомлення, що надходять як від процесорних ядер, розміщених на даному чипі, так і ззовні, визначаючи, чи спрямований запит пам'яті (або повідомлення) до якогось одного банку (або декількох банкам) DRAM, які розміщені на розглянутому чипі системи пам'яті, або до зовнішньої пам'яті, що повинен бути спрямований через інтерфейс зовнішньої пам'яті 4. Інтерфейс вводу-виводу 8 може містити як послідовні, так і паралельні комунікаційні порти. У загальному випадку кількість цих портів може бути масштабованою так, що кількість інших подібних чипів і пристроїв вводу-виводу, які можуть бути безпосередньо пов'язані з даним чипом системи пам'яті, може бути збільшена в міру необхідності. Крім того, передбачається сумісність інтерфейсу вводу-виводу 8 з іншими необхідними для спільної роботи інтерфейсами, наприклад, інтерфейсом шини вводу-виводу типу PCI, волоконнооптичного каналу, універсальної шини, і т.п. Підтримка такої сумісності покладається на контролер пам'яті 3.

Контролер пам'яті 3 забезпечує погоджену взаємодію функціональних вузлів, розміщених як на даному чипі, так і на інших, пов'язаних з ним чипах, реалізуючи наступні функції:

- керування обміном даними між чипом системи пам'яті та іншими аналогічними чипів або пристроями вводу-виводу;
- визначення спрямованості запитів до пам'яті від процесорів або процесорних ядер (до одного або декількох банкам DRAM на даному чипі, або до зовнішньої пам'яті), і передача цих запитів або до банків DRAM на даному чипі, або до зовнішньої пам'яті через зовнішній інтерфейс пам'яті 4;
- усунення непогодженості в станах пам'яті, наприклад, шляхом запуску програмного протоколу когерентності кеш;
- підтримка обміну даними між процесорними ядрами та іншими процесорами поза кристалом і пов'язаними з ними блоками пам'яті, або пристроями вводу-виводу через інтерфейс вводу/виводу 8.

При обміні даними або спільним використанням пристроїв даного чипа з підключеними до нього пристроями, контролер пам'яті 3 використовує два механізми зв'язку: один - для підтримки локальної комунікації й запитів вводу-виводу, і другий - для підтримки віддаленої комунікації й запитів вводу-виводу. Наприклад, отримані запити вводу-виводу та повідомлення даних від процесорного ядра даного чипа, призначені для пристроїв поза цим чипом, передаються до належного місця розташування цих пристроїв поза чипом.

На контролер пам'яті також можуть бути покладені функції керування повідомленням і маршрутизацією вводу-виводу. Для цього він повинен містити таблиці маршрутизації або команди маршрутизації й повинен бути запрограмований, таким чином, щоб обробляти ці функції. Крім того, на контролер пам'яті 3 можуть також покладатися функції керування завантаженням і виконанням деяких операцій з пам'яттю на чипі, зовнішньою пам'яттю, або пам'яттю, що перебуває на інших аналогічних чипах, наприклад, функції кешування й попередньої вибірки деяких операцій для реалізованої програми або операційної системи.

Паралельний інтерфейс до зовнішньої пам'яті 4, підключений безпосередньо до контролера пам'яті 3, призначений для електричного взаємозв'язку сумісними сигналами, що надходять від системи пам'яті на чипі до зовнішньої пам'яті й назад, забезпечуючи реалізацію функцій читання й записи інформації з/у зовнішній пам'яті. Як зовнішня пам'ять може використатися будь-який тип як енергонезалежного, так й енергозалежного



запам'ятовувального пристрою (наприклад, пам'ять типу DRAM). Фактично інтерфейс зовнішньої пам'яті 4 - це порт розширення пам'яті на чипі за рахунок підключення додаткової зовнішньої пам'яті.

Інтерфейс завантаження 9 призначений для завантаження в кеш команд 14 ведучих процесорів 13 і кеш команд 26 процесорних ядер 25 кожного банку пам'яті 1 (Фіг.2) програми початкового завантаження, а також для завантаження констант у пам'ять констант 32 першого селектора вибору 17 для вибору ведучих процесорів (Фіг.1, Фіг.3). Після завантаження ця програма використовується процесором, щоб запустити операційну систему й відповідну команду.

У загальному випадку завантажувальний інтерфейс припускає наявність завантажувального програмувального (напівпостійного) запам'ятовувального пристрою (ППЗП), що підтримує системну програму початкового завантаження.

Інтерфейс діагностики й налагодження 10 призначений для проведення зовнішньої експертизи станів процесорних елементів, розміщених на чипі системи пам'яті, і виконання їхнього налагодження. Для цього він містить сукупність інтерфейсних входів/виходів, підключених до групи вихідних-вхідних контактів "Діагностика/налагодження", що утворюють діагностичний інтерфейс для підключення електрично й програмно сумісних апаратів налагодження до кристала системи пам'яті.

З огляду на те, що пропонується система пам'яті може бути рекомендована до масового виробництва, тому що фактично вона є універсальною за рахунок настроювання її архітектури на клас розв'язуваних завдань, застосування інтерфейсу 10 у складі чина системи пам'яті є необхідним, тому що крім регулярного масиву пам'яті в нього вбудовані нерегулярні логічні схеми, які утрудняють верифікацію й налагодження BIC. Ця вбудована надмірність, що забезпечує спеціальну підтримку зовнішнього іспитового обладнання, дозволяє знизити витрати на тестування й налагодження BIC.

Внесення в систему пам'яті пропонованої корисної моделі нових вузлів і відповідно нових зв'язків призводить у порівнянні із прототипом до розширення функціональних можливостей вузла інтерфейсу діагностики й налагодження 10, що забезпечує можливість верифікації та налагодження як системи в цілому, так і при необхідності окремо особливо відповідальних вузлів. Це досягається за рахунок того, що тестові сигнали (сигнали налагодження) надходять із других, третіх і четвертих виходів вузла інтерфейсу 10 відповідно на треті входи першого селектора 17 вибору ведучого процесора 13, на перші входи другого селектора 18 вибору банку пам'яті 1 і банку пам'яті 2, а також на перші входи третього селектора 19 вибору підбанку пам'яті 20 (Фіг.1).

Перший селектор вибору 17. Призначений для формування й видачі керуючих сигналів для вибору одного або декількох одночасно працюючих ведучих процесорів 13 (Фіг.1, Фіг.3). Керуючі сигнали утворюють s-розрядний керуючий код, так що кожному з s використовуваних процесорів 13 відповідає один розряд s-розрядного керуючого коду. Сукупність керуючих кодів (керуючих слів) утворює масив, кожне слово якого визначає вибір оптимального по параметрах ведучого процесора для реалізації на кожній фазі відповідної гілки розпаралеленого алгоритму. Цей масив формується за допомогою вузла 5, розміщеного на чипі, або за допомогою зовнішньої, стосовно чипа, ЕОМ (наприклад, HOST-машиною) як перед реалізацією алгоритму, так й у процесі функціонування системи пам'яті за допомогою відповідного компілятора. Оскільки під час рішення заздалегідь сформованого фрагмента алгоритму набір керуючих слів не змінюється й над цими словами не здійснюється яких-небудь перетворень, цей набір визначений як набір констант, що розміщається в пам'яті констант 32 першого селектора вибору 17 (Фіг.3). Вибір у належний час потрібної константи здійснюється дешифратором констант 31, на входи якого з виходів лічильника фаз 30 надходить код, що фіксує закінчення поточної фази гілки реалізованого алгоритму. При цьому на вхід лічильника 30 з виходів вузла 5 надходить послідовність сигналів - ознак закінчення фаз. Приклад формування послідовності ознак фаз і наборів констант верхнього рівня (BP) для вибору ведучих процесорів  $BP_s$  ( $s=1, 2, 3, 4$ ), а також ознак фаз і наборів констант середнього рівня (CP) для вибору банків пам'яті  $m_i$  ( $i=1, 2, 3, 4$ ) (розподіл підбанків пам'яті тут не приймається в увагу) наведений на Фіг.5. Всі процесори  $BP_s$  і банки пам'яті  $m_i$  розподілені по паралельних гілках A, B, C, D алгоритму розв'язуваного завдання, при цьому порядковий номер позначених на малюнку процесорів  $BP_1, BP_2, BP_3, BP_4$  збігається з порядковим номером (зліва праворуч) коду константи ознак фаз BP незалежно від розміщення їх по паралельних гілках A, B, C, D. Аналогічно порядковий номер позначених на малюнку банків пам'яті  $m_1, m_2, m_3, m_4$  також збігається з порядковим номером (зліва праворуч) коду константи ознак фаз CU незалежно від розміщення їх по паралельних гілках A, B, C, D і ведучим процесорам  $BP_s$ . Наприклад, код ознаки фази BP, рівний 0110, означає, що в момент  $T_3$  закінчення третьої фази на наступній четвертій фазі (Фіг.5) ініціалізуються лише процесори  $BP_s$  (паралельна гілка B) і  $BP_3$  (паралельна гілка C), а по закінченні четвертої фази (у момент  $T_4$ ) додатково включається в роботу на паралельній гілці D процесор  $BP_4$  (код 0111). Отримана вузлом 5 послідовність зафіксованих моментів (тактів) закінчення фаз у вигляді  $T_1, T_2, \dots, T_9$  надходить на вхід лічильника фаз 30 (Фіг.3), переводячи його щораз у новий стан, код з виходів якого є кодом адреси для вибору відповідної константи, розміщеної в пам'яті констант 32. Отриманий у такий спосіб код константи з пам'яті констант, надходить на відповідні перші "одиночні" входи регістра 33, устанавлюючи в стан "одиниця" розряди, номери яких відповідають одноименним номерам ведучих процесорів, що підлягають ініціалізації на наступній фазі. При діагностиці та налагодженні системи аналогічні за структурою коди від інтерфейсу налагодження 10 надходять на другі "одиночні" входи регістра 33, здійснюючи вибір ведучих процесорів для діагностики й відладки.

Необхідний набір констант може бути завантажений з виходів вузла 5 через відповідні входи групи перших входів селектора (Фіг.1, Фіг.3), а також при необхідності - через другі входи селектора від зовнішніх ЕОМ, або інших приєднаних чипів через інтерфейс завантаження 9 (Фіг.1, Фіг.3).

Другий селектор вибору 18. Структура та організація роботи другого селектора вибору 18 ідентичні структурі та організації роботи першого селектора вибору 17, які описані вище (Фіг.3, Фіг.4). Відмінність у загальному випадку може складатися лише в тім, що кількість розрядів констант фаз середнього рівня (CP) не дорівнює кількості розрядів констант фаз верхнього рівня (BP), оскільки кількість ведучих процесорів (наприклад, усього два) може відрізнятися від кількості обраних банків пам'яті (наприклад, шістнадцять). Точно також послідовності ознак фаз BP та ознак фаз CP можуть відрізнятися, наприклад, як показано на Фіг.5. У зв'язку із цим розрядність констант фаз і розрядність пам'яті констант 32 другого селектора вибору 18, а також відповідно розрядність регістра 33 цього селектора (Фіг.4) також будуть відрізнятися від кількості розрядів констант, що містяться в пам'яті констант фаз 32 і кількості розрядів регістра 33 першого селектора 17 (Фіг.3, Фіг.4).

Третій селектор вибору 19. Структура та організація роботи третього селектора вибору 19 ідентичні структурі та організації роботи другого селектора 18, які описані вище (Фіг.4). Відмінність може складатися лише в тім, що кількість розрядів констант фаз нижнього рівня (НР) не дорівнює кількості розрядів констант фаз середнього рівня (СР), оскільки кількість банків пам'яті (наприклад, шістнадцять) може відрізнятися від кількості обраних підбанків пам'яті (наприклад, чотири). Точно також і послідовності ознак фаз НР й ознак фаз СР можуть відрізнятися. У зв'язку із цим розрядність констант фаз, розрядність пам'яті констант 32, а також відповідно розрядність регістра 33 третього селектора вибору також будуть відрізнятися від кількості розрядів констант, що містяться в пам'яті, фаз 32 і кількості розрядів регістра 33 другого селектора 18 (Фіг.1, Фіг.4).

Пропонована система пам'яті з інтеграцією функцій зберігання й обробки інформації на одному кристалі може працювати в двох основних режимах:

1). В якості звичайного запам'ятовуючого пристрою (ЗП) з реалізацією функцій запису, зберігання й видачі інформації ("Режим ЗП"), що може бути використане як доповнення до основної пам'яті процесора, розташованого поза даним кристалом, або як кеш-пам'ять різного призначення.

2). Як середовище обробки (СО) і зберігання інформації ("Режим СО"), що може бути використано як приставка - прискорювач для ЕОМ, кластера, або комп'ютерної системи іншої конфігурації для підвищення продуктивності при виконанні масових операцій з пам'яттю з можливістю глибокого розпаралелювання (наприклад, операцій з пікселями при обробці зображень, при обробці даних радарних сигналів, сигналів медичних гістограм і т.п.).

У цих режимах пропонується система пам'яті може виконувати два типи операцій: операції типу А та операції типу В. Операції типу А реалізуються усередині кожного підбанку пам'яті із застосуванням комутатора 21 для зв'язку між регістрами й процесорними ядрами різних підбанків на базі внутрішніх ресурсів банку без виходу за його межі та за межі чипа до інших пристроїв за допомогою наявних інтерфейсних схем. Операції типу В реалізуються на базі ресурсів наявних банків пам'яті з використанням комутації між банками за допомогою загального для них регістра рядка 11 і спеціальних схем: другого селектора 18 вибору банку пам'яті 1 і банку пам'яті 2, третього селектора 19 вибору підбанку пам'яті 20, селектора читання 6 і декодера запису 7. При цьому використовуються наявні на чипі інтерфейсні схеми для видачі інформації за межі чипа іншим пристроям (наприклад, зовнішньої пам'яті), а також для одержання інформації від них. Перехід від одного типу операцій до іншого здійснюється за допомогою сигналів "Тип А", "Тип В", які надходять із п'ятих виходів контролера 3 на п'яті входи банків пам'яті (Фіг.1).

У режимі ЗП пропонується система пам'яті реалізує функцію зберігання інформації та набір операцій "Читання", "Запис", характерних для оперативної пам'яті комп'ютерних систем. Однак, на відміну від виконання цих операцій у класичній пам'яті комп'ютерних систем і від системи пам'яті - прототипу, у пропонованій системі пам'яті при виконанні цих операцій є можливість зчитувати та записувати як повне N-розрядне слово, що містить від декількох сотень до декількох тисяч розрядів (біт), так і фрагменти цього слова. При цьому розрядність фрагмента може бути кратна величині R - розрядності одного банку пам'яті 1, або кратна величині r - розрядності підбанку пам'яті 20. Це досягається за рахунок уведених другого 18 і третього 19 селекторів вибору (Фіг.1), а також регістрів підбанку 27 із входньою логікою 28 і вихідною логікою 29 (Фіг.2), за допомогою яких або дозволяється, або забороняється функціонування відповідних банків і підбанків пам'яті та передача інформації усередині кристала, а також із входу на вихід кристала й назад під керуванням відповідних полів пакета, що надходить на перші входи вузла 5. Тому будь-яка операція в пропонованій системі пам'яті починається з аналізу полів пакета вузлом 5, що надходить на його перші входи або від HOST - машини, або від інших аналогічних чипів, і формування необхідних для виконання операції керуючих сигналів.

Розглянемо спочатку в режимі ЗП процедуру виконання операції читання з масивів пам'яті 22 повного N-розрядного рядка ("Читання 1/ЗП"). При цьому виділимо дві фази виконання цієї операції: на першій фазі відбувається зчитування інформації з масивів пам'яті 22 (Фіг.2) і занесення її на регістр рядка 11 (ф.1\_Читання 1/ЗП), на другій фазі (ф.2\_Читання 1/ЗП) інформація, занесена на регістр рядка 11, передається на вихід кристала системи пам'яті або на зовнішню пам'ять через інтерфейс зовнішньої пам'яті 4, або до інших аналогічних чипів, або до інших зовнішніх пристроїв через інтерфейс вводу-виводу 8.

Операція читання в режимі ЗП, як і будь-які інші операції системи пам'яті, починається з аналізу полів керуючого пакета, що надійшов на перші входи вузла 5, формування та видачі вузлом 5 керуючих сигналів і сигналів синхронізації на відповідні входи вузлів і схем системи пам'яті. Одночасно з надходженням керуючого пакета на входи вузла 5, на входи "Адреса рядка" чипа системи пам'яті надходить код адреси рядка, що через шості входи банків пам'яті 1 і 2 надходить на перші входи дешифраторів 23 підбанків пам'яті 20 (Фіг.2). При цьому на відповідні другі входи всіх банків пам'яті з виходів другого селектора вибору 18 надходить (m+1) розрядне керуюче слово (m - кількість банків пам'яті 1 плюс один банк контрольних кодів 2), кожен розряд якого дорівнює "одиниці", тобто дозволене функціонування всіх банків пам'яті. На відповідні треті входи банків пам'яті з виходів третього селектора вибору 19 надходить q-розрядне слово (q - кількість підбанків пам'яті 20 одного банку пам'яті), кожен розряд якого також дорівнює "одиниці", тобто дозволена робота всіх підбанків пам'яті 20 кожного банку пам'яті.

Одночасно з кодом адреси рядка через вхід "Ввід/вивід" кристала системи пам'яті на третій вхід інтерфейсу вводу-виводу 8 приходить запит на читання інформації. Інтерфейс вводу-виводу з першого свого виходу передає інформацію про запит читання на другий вхід контролера пам'яті 3, що визначає, куди повинен бути спрямований цей запит, а також результат запиту: або до банків пам'яті даного кристала, або транзитом до інших аналогічних чипів через інтерфейс вводу-виводу 8, або до зовнішньої пам'яті через інтерфейс зовнішньої пам'яті 4 (Фіг.1).

При реалізації режиму "Читання 1/ЗП" запит спрямовано до внутрішньої пам'яті типу DRAM, що розміщена на даному кристалі, а результат запиту (зчитане слово) - до інших аналогічних чипів через інтерфейс вводу-виводу 8, або до зовнішньої пам'яті через інтерфейс зовнішньої пам'яті 4 (Фіг.1). Тому контролер пам'яті 3 під впливом керуючих сигналів, що надходять на його четверті входи з відповідних сьомих виходів вузла 5, формує діаграму погодженої взаємодії функціональних вузлів, пов'язаних з пам'яттю розміщених на даному чипі, а також сигнал "Читання" (наприклад, високий рівень сигналу "Читання/Запис"), що із четвертого виходу цього контролера надходить на однойменні четверті входи всіх банків пам'яті й далі - на другі входи дешифраторів 23 (Фіг.2). При цьому на третьому вході 3 дешифраторів всіх банків пам'яті встановлюється розв'язний сигнал - логічна "одиниця", що через другий вхід банку пам'яті надходить із першого виходу другого селектора вибору 18.

Дешифратор кожного банку пам'яті (Фіг.2) при наявності сигналу "Читання" і сигналу логічної "одиниці" на його третьому вході, а також, що надійшов на перші входи коду адреси рядка видає на вихід сигнал вибірки, що надходить на вхід обраного відповідно до коду адреси рядка масиву пам'яті 22 всіх підбанків пам'яті 20, зчитуючи інформацію із всіх однойменних обраних рядків всіх банків пам'яті одночасно, щоб створити рядок даних розрядністю N. Кожен розряд цього рядка даних через буферну схему 24 і другий вхід вхідної логіки 28 (Фіг.2) при наявності на її четвертому вході розв'язного сигналу "Тип В" заноситься на відповідну розрядну позицію регістра 27. При цьому на його п'ятому вході присутній сигнал дозволу - логічна "одиниця", що надходить через треті входи банку пам'яті із других виходів третього селектора вибору 19. Далі цей сигнал через третій вихід вхідної логіки 29 цього регістра (при наявності сигналу дозволу "Тип В" на четвертому вході вхідної логіки 29) надходить на відповідні другі входи банку пам'яті. Сукупність зчитаних розрядів всіх банків пам'яті передається потім на відповідні другі входи схеми виявлення й виправлення помилок 12, що виконує свої функції лише при наявності на її третьому вході керуючого сигналу, що надходить із дев'ятого виходу вузла 5. Проаналізований (і при необхідності відкоректований) N-розрядний рядок з перших виходів схеми виявлення й виправлення помилок 12 разом з контрольними розрядами передається на регістр рядка 11, де тимчасово запам'ятовується (Фіг.1, Фіг.2). Цим закінчується перша фаза (ф.1\_Читання 1/3П).

У другій фазі (ф.2\_Читання 1/3П) інформацію, що перебуває на регістрі 11, необхідно передати за межі кристала, наприклад, на зовнішню пам'ять або на інші пристрої. Для цього, насамперед, з усього N-розрядного слова необхідно виділити потрібне слово необхідної розрядності, що відповідає б розрядності пристрою, куди це слово передається. Тому одночасно з кодом адреси рядка через входи "Адреса стовпця" кристала на треті входи селектора читання 6 надходить код адреси стовпця, а на четверті його входи - ознака режиму читання (видача на вихід декількох k-розрядних слів або одного k-розрядного слова,  $k \leq N$ ). По ознаці режиму та коду адреси стовпця селектор читання виділяє на регістрі рядка 11 слово відповідної розрядності, використовуючи зв'язок: перші входи й перші входи селектора читання з'єднані відповідно із другими входами й першими виходами регістра рядка. Виділене слово із других виходів селектора читання (Фіг.1) надходить на другі входи інтерфейсу вводу-виводу 8 для видачі цього слова на вихід через треті входи чипа (при наявності на четвертому вході інтерфейсу 8 розв'язного сигналу з п'ятого виходу вузла 5) або на другі входи інтерфейсу зовнішньої пам'яті 4 для видачі зчитаного слова на виходи чипа "Зовнішня пам'ять" (при наявності на його четвертому вході сигналу із шостого виходу вузла 5).

Читання з масиву пам'яті 22 замість довгого N-розрядного рядка слова потрібної розрядності (Читання 2/3П) для передачі його за межі чипа системи пам'яті виконується як і при реалізації операції (Читання 1/3П), за винятком того, що логічна "одиниця" з виходів другого селектора вибору 18 надходить на відповідні другі входи тільки обраних банків пам'яті, де це слово розміщене, тобто дозволене функціонування тільки обраних банків пам'яті, а робота інших банків пам'яті блокується, тому що відповідні розряди керуючого слова, що надходить із виходів другого селектора вибору, рівні "нулю". Аналогічно логічна "одиниця" з виходів третього селектора вибору 19 надходить на відповідні треті входи й далі на п'яті входи регістрів 27 і четверті входи процесорних ядер 25 тільки обраних підбанків пам'яті 20 обраних банків, де це слово розміщене, тобто дозволене функціонування тільки обраних підбанків пам'яті, а робота інших підбанків пам'яті блокується, тому що відповідні розряди керуючого слова, що надходить із виходів третього селектора вибору, рівні "нулю". При цьому, як і при реалізації операції (Читання 1/3П), операція (Читання 2/3П) також виконується за дві фази (ф.1\_Читання 2/3П) і (ф.2\_Читання 2/3П).

Можлива також реалізація операції (Читання 3/3П) як модифікація операції (Читання 1/3П), коли керуючий сигнал, що надходить із дев'ятого виходу вузла 5 на третій вхід схеми виявлення й виправлення помилок має значення "нуль". При цьому робота схеми 12 блокується, і вихідні сигнали, що надходять із банків пам'яті 1 та 2 на другі входи схеми виявлення й виправлення помилок 12, проходять транзитом на відповідні входи регістра рядка 11, без істотних затримок сигналів цією схемою. При цьому також виділяються дві аналогічні фази: (ф.1\_Читання 3/3П) і (ф.2\_Читання 3/3П).

У такий же спосіб операція (Читання 4/3П) як модифікація операції (Читання 2/3П) при відключеній схемі виявлення й виправлення помилок 12 може бути виконана за дві фази (ф. 1\_Читання 4/3П) і (ф.2\_Читання 4/3П).

При виконанні операції запису в режимі 3П ("Запис/3П") реалізується запис інформації (дані або програми) у масиви пам'яті 22 банків пам'яті 1 та 2 (Фіг.2), що надходить або із зовнішньої пам'яті через паралельний інтерфейс зовнішньої пам'яті 4 (Фіг.1), або із пристроїв вводу-виводу або аналогічних чипів через інтерфейс вводу-виводу 8 (Фіг.1). У першому випадку k-розрядне слово надходить через контакти "Зовнішня пам'ять" кристала на треті входи інтерфейсу зовнішньої пам'яті 4, на четверті входи якого із шостих виходів вузла 5 надходить керуюча інформація, що визначає режим запису (Фіг.1). Одночасно з першого виходу інтерфейсу зовнішньої пам'яті 4 на перші входи контролера пам'яті 3 надходить інформація, по якій контролер пам'яті при наявності на його четвертому вході керуючої інформації із сьомих виходів вузла 5 визначає, чи направляється дане слово для запису в масив пам'яті даного кристала або для передачі транзитом на інші кристали. При реалізації режиму "Запис/3П" інформація передається для запису в банки пам'яті 1 та 2. Тому із четвертих виходів контролера 3 на четверті входи всіх банків пам'яті спочатку надходить сигнал "Читання" команди "Читання/Запис" (наприклад, сигнал "Читання" - високий рівень, а "Запис" - низький), а через входи "Адреса рядка" кристала й шості входи банків пам'яті на перші входи дешифраторів 23 банків пам'яті 1 та 2 надходить код адреси рядка. Далі виконується перша фаза однієї з модифікацій розглянутих вище операцій читання в режимі 3П.

Розглянемо спочатку виконання операції запису інформації ("Запис 1/3П") коли попереднє перед записом читання виконується в режимі (ф.1\_Читання 1/3П), докладно описаного вище. При цьому в результаті виконання першої фази читання зчитаний N-розрядний рядок разом з контрольними розрядами розміщено в регістрі рядка 11 (Фіг.1, Фіг.2).

Одночасно з ініціалізацією контролера 3 інтерфейс зовнішньої пам'яті 4 видає на другі входи k-розрядне слово та відповідну ознаку режиму запису, які надходять на перші входи декодера запису 7. При цьому на треті входи декодера запису через вхідні контакти кристала "Адреса стовпця" надходить код адреси стовпця, а на п'яті входи декодера запису надходить із перших виходів третього селектора вибору 19 код дозволу, що разом з ознакою режиму й кодом адреси стовпця забезпечує можливість декодеру стовпця 7 визначити місце розташування вхідного k-розрядного слова на довгому N-розрядному рядку ( $k \leq N$ ) регістра рядка 11, заміщаючи (використовуючи перший вихід декодера запису і перший вхід регістра рядка) у виділеній області довгого слова

наявну інформацію на записуване k-розрядне слово. Тому що зміст регістра рядка 11 після такої заміни змінився, то новий N-розрядний рядок знову піддається аналізу (і при необхідності корекції) з боку схеми аналізу й виправленні помилок 12. Інформаційні розряди відкоректованого рядка із четвертих виходів регістра рядка 11 надходять через сьомі входи банків пам'яті 1 (а контрольні розряди - відповідно з п'ятих виходів регістра 11 на сьомі входи банку пам'яті 2) на однойменні розрядні позиції п'ятих виходів вхідної логіки 28 для запису в кожен регістр 27 при наявності керуючого сигналу "Тип В" відповідно на п'ятих керуючих входах цих регістрів. Далі інформація з перших виходів кожної вихідної логіки 29 (Фіг.2) регістра 27 (при наявності сигналу дозволу "Тип В" на четвертому вході вихідної логіки) надходить на перші входи буферної схеми 24 і далі з її виходів на однойменні розрядні входи масиву пам'яті 22. Одночасно із четвертих виходів контролера пам'яті 3 (Фіг.1) через четверті входи банків пам'яті 1 та 2 на другі входи дешифраторів 23 всіх банків пам'яті (Фіг.2) надходить сигнал "Запис", при цьому код адреси рядка на перших входах дешифраторів 23 не змінюється. Не змінюється й логічна "одиниця" на третіх входах цих дешифраторів. Відбувається запис оновленого рядка на те ж саме місце, звідки був тільки що зчитано рядок та у який була зроблена запис k-розрядного слова, що надійшло від джерел поза кристалом. Таким чином, операцію запису в режимі 3П ("Запис 1/3П") можна також розділити на дві фази: перша фаза - попереднє читання на регістр 11 N-розрядного слова (ф. 1\_Запис 1/3П), і друга фаза - вибір області на довгому N-розрядному рядку, у якій повинне бути внесене k-розрядне слово, підлягаюче запису ( $k < N$ ), і безпосередньо запис нового N-розрядного слова в масиви пам'яті 22 всіх банків пам'яті (ф.2\_Запис 1/3П). Залежно від того, який різновид першої фази використовується при попереднім читанні в циклі запису (ф.1\_Читання 1/3П або ф.1\_Читання 2/3П, або ф.1\_Читання 3/3П, або ф.1\_Читання 4/3П), можна відповідно виділити кілька модифікацій операції запису: "Запис 1/3П", "Запис 2/3П", "Запис 3/3П" та "Запис 4/3П". При цьому друга фаза кожної модифікації операції запису реалізується однаково.

У пропонованій системі пам'яті на відміну від системи-прототипу можлива реалізація операції запису по скороченому циклі ("Запис 5/3П"). У цьому випадку фаза попереднього читання відсутня, а k-розрядне слово на запис за допомогою декодера запису заноситься на відповідні розрядні позиції регістра 11, і далі із четвертих виходів цього регістра k-розрядне слово через сьомі входи банків пам'яті надходить на регістри 27 і потім записується в масиви пам'яті 22 тільки обраних за допомогою другого селектора 18 і третього селектора 19 відповідних банків пам'яті 1 і підбанків пам'яті 20. При цьому адреса стовпця на третіх входах декодера запису 7 та адреса рядка на шостих входах банку пам'яті 1 і далі на перших входах дешифраторів 23 виставляється одночасно з надходженням на другий вхід дешифратора сигналу "Запис", при цьому спрацювають дешифратори тільки обраних підбанків пам'яті, на третіх входах яких є присутнім сигнал дозволу - логічна "одиниця", що надходить із відповідного першого виходу другого селектора вибору 18 (Фіг.1, Фіг.2).

При роботі пропонованої системи пам'яті в режимі СО на перші входи вузла інтерфейсів й обробки керуючого пакета 5 надходить керуючий пакет, що містить поля різного призначення, зазначені вище при описі вузла 5, які необхідні для функціонування в цьому режимі системи в цілому та забезпечення зв'язку даного чипа з іншими аналогічними чипами, а також із зовнішньою пам'яттю та іншими зовнішніми пристроями. При цьому вузол 5 видає на виходи 1, 3 та 2 послідовності ознак фаз роботи ведучих процесорів 13, банків 1 та 2, а також підбанків пам'яті 20 при розподілі їх по паралельних гілках алгоритму відповідно до карти розподілу ресурсів, приклад якої наведений на Фіг.5. Така карта може бути введена у вузол 5 ззовні (наприклад, HOST-машинною), або створена самим вузлом 5 по даним керуючого пакета. Згідно із цими послідовностями селектори вибору 17, 18 та 19 (Фіг.3, Фіг.4) визначають при реалізації паралельних гілок алгоритму проміжки часу функціонування ведучих процесорів 13, банків пам'яті 1 та 2, підбанків пам'яті 20 і відповідно процесорних ядер 25 кожного підбанку й видають відповідні сигнали керування на перші входи ведучих процесорів, а також на другі та на треті входи банків пам'яті. Тому реалізація будь-яких операцій системою пам'яті в режимі СО розглядається саме в моменти функціонування зазначених компонентів, припускаючи при цьому, що є можливість настроювання (або перебудови) архітектури системи на різні формати даних та особливості реалізованого алгоритму.

Тому що основою середовища обробки інформації такого чипа є пам'ять, то перш ніж описувати організацію обробки інформації в цьому середовищі, розглянемо особливості виконання в режимі СО базових операцій запису в пам'ять ("Запис/СО"), читання інформації з пам'яті ("Читання/СО") і операції пересилання ("Пересилання/СО") при доставці потрібних для виконання операцій даних до місця обробки та відправлення отриманих результатів операцій від засобів обробки до місця їхнього призначення. Є кілька різновидів цих операцій залежно від вузла, що робить запит на одержання та відправлення даних, і місця їхнього добування при читанні, а також місця призначення передачі даних для подальшого їхнього використання. У зв'язку із цим операції запису та читання разом з відповідними операціями пересилання даних будемо вважати макроопераціями, виконання яких підтримується апаратними засобами.

До макрооперацій запису інформації в пам'ять в режимі СО належать:

"Запис 1/СО" реалізується по запиту ведучих процесорів 13 для запису результатів обробки інформації цими процесорами, які надходять із перших виходів (Фіг.1) через перші входи вузла 16 у кеш-пам'ять даних 15. При цьому запис у поточний момент часу здійснюється в кеш-пам'ять тих процесорів, які ініціалізовані першим селектором вибору 17 відповідно до коду константи вибору, що видає селектор 17 відповідно до карти розподілу ресурсів.

"Запис 2/СО" реалізується по запиту ведучих процесорів 13 для запису вмісту регістра рядка 11 через селектор читання 6 у кеш-пам'ять даних 15 вузла 16 (Фіг.1). При цьому дані в регістр рядка 11 надходять як з банків пам'яті 1 та 2 при виконанні першої фази операції читання (ф.1\_Читання /3П), так і від інших пристроїв поза кристалом при виконанні операції запису (ф.1\_Запис /3П) у різних їхніх модифікаціях. Селектор читання 6 під впливом керуючих сигналів вибору підбанку пам'яті на четвертих входах селектора, коду адреси стовпця на третіх входах селектора виділяє з довгого N-розрядного рядка, записаного раніше в регістр рядка 11, k-розрядне слово для запису його в кеш-пам'ять даних 15 відповідної розрядності. Можливо також наявність у регістрі рядка не N-розрядного слова, а k-розрядного слова, що було записано на регістр 11 при виконанні різних модифікацій операцій читання й запису: ф.1\_Читання 2/3П, ф.1\_Читання 3/3П, ф.1\_Читання 4/3П, ф.1\_Запис 2/3П, ф.1\_Запис 3/3П, ф.1\_Запис 4/3П. При цьому процес запису цього слова залишається таким же.

"Запис 3/СО" реалізується по запиту процесорних ядер 25 при записі результатів обробки інформації процесорними ядрами в масиви пам'яті 22 через регістри 27 і через буферні схеми 24 (Фіг.2). Ініціалізований процесорним ядром 25 запит із другого його виходу через перший вихід банку пам'яті (Фіг.2) надходить на третій

вхід контролера пам'яті 3, що при наявності керуючого сигналу на четвертому вході із сьомого виходу вузла 5 визначає, чи спрямований цей запит у банки пам'яті чипа, або за його межі: до зовнішньої пам'яті або до інших зовнішніх пристроїв. У цьому випадку запит спрямований у банки пам'яті, тому контролер пам'яті 3 із четвертого виходу видає сигнал "Запис", а з п'ятого виходу - розв'язний сигнал "Тип А" (Фіг.1), що через відповідні п'яті входи банків пам'яті надходить на треті входи вхідної логіки 28, а також на треті входи вихідної логіки 29 регістрів 27 (Фіг.2). Сигнал "Запис" надходить через четверті входи банків пам'яті 1 та 2 на другі входи дешифраторів 23, при цьому на треті входи дешифраторів надходить сигнал дозволу з виходів другого селектора 18 тільки для обраних відповідно до карти розподілу ресурсів банків пам'яті, а на п'яті входи регістра 27 - сигнал дозволу тільки для обраних підбанків пам'яті із других виходів третього селектора 19. При наявності перерахованих керуючих сигналів результат виконання операції процесорним ядром, представлений у вигляді k-розрядного слова, із третіх його виходів через перші входи вхідної логіки 28 запам'ятовується на регістрі 27 і далі з перших виходів вихідної логіки 29 надходить на другі входи буферної схеми 24. Одночасно з надходженням на другий вхід контролера 3 сигналу запита, на входи кристала пам'яті "Адреса рядка" надходить код рядка, що проходить на шості входи банків пам'яті й далі - на перші входи дешифраторів 23. Дешифратор відповідно до коду адреси вибирає рядок масиву 22, у яку й відбувається запис k-розрядного слова через буферну схему 24 обраного підбанку пам'яті 20. Аналогічні операції запису результатів можуть бути виконані паралельно для всіх процесорних ядер 25 всіх підбанків пам'яті.

До макрооперацій читання інформації з пам'яті в режимі СО належать:

"Читання 1/СО" реалізується по запиту ведучих процесорів 13 для читання з кеш-пам'яті даних 15 вузла 16 на регістр рядка 11 з метою подальшого використання цієї інформації при реалізації арифметичних і логічних операцій, а також її запису в банки пам'яті 1 або передачі за межі чипа. Інформація з виходів кеш-пам'яті даних 15 через другі входи вузла 16 надходить на четверті входи декодера запису 7, на треті входи якого надходить код адреси стовпця, а на п'яті входи - сформований третім селектором 19 розв'язний сигнал, що дозволяє визначити на регістрі рядка розрядністю N область для розміщення зчитаного з кеш-пам'яті k-розрядного слова та записати його в регістр рядка 11.

"Читання 2/СО" реалізується по запиту процесорних ядер 25 для читання даних з масиву пам'яті 22 підбанків пам'яті 20 на регістр рядка підбанку 27 з метою подальшого використання цієї інформації при реалізації арифметичних і логічних операцій процесорними ядрами 25 і ведучими процесорами 13, а також для її передачі за межі чипа (Фіг.2, Фіг.1).

Ініціалізований процесорним ядром 25 запит із другого його виходу через перший вихід банку пам'яті (Фіг.2) надходить на третій вхід контролера пам'яті 3, що при наявності керуючого сигналу на четвертому вході із сьомого виходу вузла 5 визначає, чи спрямований цей запит у банки пам'яті чипа, або за його межі: до зовнішньої пам'яті або до інших зовнішніх пристроїв. У цьому випадку, запит спрямований у банки пам'яті, тому контролер пам'яті 3 із четвертого виходу видає сигнал "Читання", а з п'ятого виходу - розв'язний сигнал "Тип А" (Фіг.1), що через відповідні п'яті входи банку пам'яті надходить на третій вхід вхідної логіки 29 регістра 27 (Фіг.2). Сигнал "Читання" далі надходить через четвертий вхід банку пам'яті на другі входи дешифраторів 23, при цьому на треті входи дешифратора надходить сигнал дозволу з перших виходів другого селектора 18 тільки для обраних відповідно до карти розподілу ресурсів банків пам'яті, а на п'ятий вхід регістра 27 - сигнал дозволу тільки для обраних підбанків пам'яті із другого виходу третього селектора 19. Дешифратор обраного банку пам'яті (Фіг.2) при наявності сигналу "Читання" і сигналу логічної "одиниці" на його третьому вході, а також, що надійшов на перші входи коду адреси рядка видає на свій вихід сигнал вибірки, що надходить на вхід обраної відповідно до коду адреси рядка масиву пам'яті 22, зчитуючи інформацію з однойменних обраних рядків обраних підбанків пам'яті одночасно, що створює рядок даних необхідної розрядності. Кожен розряд цього рядка даних через буферну схему 24 і вхідну логіку 27 (Фіг.2) при наявності на її третьому вході розв'язного сигналу "Тип А" заноситься на відповідну розрядну позицію регістра 27 (при наявності на його п'ятому вході дозволу логічної "одиниці" - сигналу вибору підбанку пам'яті).

"Читання 3/СО" реалізується по запиту процесорних ядер 25 при читанні даних із зовнішньої пам'яті на регістр рядка 11 через інтерфейс зовнішньої пам'яті 4 або з інших зовнішніх пристроїв через інтерфейс вводу-виводу 8 і декодер запису 7 (Фіг.2, Фіг.1).

Ініціалізований процесорним ядром 25 запит з його другого виходу через перший вихід банку пам'яті (Фіг.2) надходить на третій вхід контролера пам'яті 3, що при наявності керуючого сигналу на четвертому вході із сьомого виходу вузла 5 визначає, чи спрямований цей запит у банки пам'яті чипа, або за його межі: до зовнішньої пам'яті або до інших зовнішніх пристроїв. У цьому випадку запит спрямований до зовнішньої пам'яті, тоді контролер пам'яті 3 виробляє сигнали запиту до зовнішньої пам'яті, що разом із сигналами синхронізації надходить із перших виходів контролера на перші входи інтерфейсу зовнішньої пам'яті 4. Якщо зовнішня пам'ять у момент звертання до неї виявляється зайнятою, тоді контролер 3 виробляє сигнали, які або переводять пам'ять в інший стан, або сигнали звернення не спрацьовуються, поки зовнішня пам'ять не звільниться. При відпрацьовуванні звернення зовнішня пам'ять видає через входи системи "Зовнішня пам'ять" на треті входи інтерфейсу 4 k-розрядне слово, що через другий вихід інтерфейсу 4 надходить на перші входи декодера запису 7. При цьому на треті входи декодера запису 7 через вхідні контакти кристала "Адреса стовпця" надходить код адреси стовпця, а на п'яті входи декодера запису подається з першого виходу третього селектора вибору 19 код дозволу, що разом з ознакою режиму та кодом адреси стовпця забезпечує можливість декодеру стовпця 7 визначити місце розташування вхідного k-розрядного слова на довгому N-розрядному рядку ( $k \ll N$ ) регістра рядка 11, замінюючи (по цепі перший вихід декодера 7 - перший вхід регістра рядка 11) у виділеній області довгого слова наявну інформацію на записуване k-розрядне слово. Тому що зміст регістра рядка 11 змінюється, то отриманий рядок піддається аналізу (і при необхідності корекції) з боку схеми аналізу й виправлення помилок 12, використовуючи для цього відповідні зв'язки: перші входи та перші входи схеми 12 - другі входи й треті входи регістра рядка 11.

Макрооперації пересилання. Є кілька різновидів макрооперацій пересилань типу регістр-регістр, регістр-процесор (процесорне ядро), виділених окремо від перерахованих вище макрооперацій, реалізація яких необхідна при виконанні арифметичних і логічних операцій у системі пам'яті, що працює в режимі СО. Основними з них є:

"Пересилання 1/CO" застосовується для реалізації пересилань вмісту регістрів 27 кожного підбанку пам'яті через другі виходи банків пам'яті 1 та 2 на регістр рядка 11 через схему виявлення й виправлення помилок 12 під керуванням сигналів "Тип В" на четвертому вході вхідної логіки 28 і на четвертому вході вихідної логіки 29 регістра 27, а також сигналів вибору банків пам'яті, що надходять із виходів другого селектора 18 на треті входи банку пам'яті й сигналів вибору підбанків пам'яті, що надходять із других виходів третього селектора 19 на треті входи банків пам'яті (Фіг.1, Фіг.2).

"Пересилання 2/CO" застосовується для реалізації пересилань вмісту регістра рядка 11 на регістри 27 кожного підбанку пам'яті та прямо на процесорні ядра 25. Таке пересилання виконується шляхом передачі інформації із четвертими виходів регістра рядка 11 на сьомі входи банків пам'яті 1 і з п'ятих виходів контрольних розрядів регістра рядка 11 на сьомі входи банку пам'яті 2. Далі ця інформація надходить на п'яті входи вхідної логіки 28 регістра 27 і на треті входи процесорного ядра 25 під керуванням сигналів "Тип В" на четвертому вході вхідної логіки 28 регістра 27, а також сигналів вибору підбанків пам'яті, що надходять із других виходів третього селектора 19.

"Пересилання 3/CO" застосовується для реалізації пересилань вмісту регістра 27 через другі виходи вихідної логіки 28 кожного підбанку пам'яті на входи комутатора 21 і з виходу комутатора 21 - на другі входи процесорного ядра 25 під керуванням сигналів "Тип А" на третьому вході вихідної логіки 29 і керуючого сигналу на третьому вході комутатора 21, а також сигналів вибору підбанків пам'яті, що надходять із виходів третього селектора 19 через треті входи підбанків пам'яті на п'ятий вхід регістра 27 і четвертий вхід процесорного ядра 25.

"Пересилання 4/CO" застосовується для реалізації пересилань результатів виконання операцій процесорними ядрами 25 на регістри 27 підбанків пам'яті під керуванням сигналів "Тип А" на третьому вході вхідної логіки 28 регістра 27, а також сигналів вибору підбанків пам'яті, що надходять із виходів третього селектора 19 через треті входи банків пам'яті на п'ятий вхід регістра 27 і третій вхід процесорного ядра 25. Інші операції (макрооперації) при роботі з пам'яттю, у значній мірі збігаються з наведеними вище, тому на них зупинятися не має змісту. Там, де це буде потрібно, вони будуть розглянуті по ходу викладу роботи системи.

Робота пропонованої системи пам'яті відбувається в такий спосіб.

Перед початком роботи необхідно здійснити завантаження системи та виконати вихідне налаштування її архітектури.

Завантаження системи. Для завантаження системи через входи "Керуючий пакет" системи пам'яті на перші входи вузла інтерфейсів й обробки пакета 5 надходить керуючий пакет, що містить поля та ознаки, а також атрибутивні та інші дані (зокрема константи фаз верхнього, середнього та нижнього рівнів вибору), необхідні для початкового налаштування архітектури системи пам'яті на клас розв'язуваних завдань та особливості рішення поточного завдання (Фіг.1). Вузол 5 аналізує поля пакета й видає відповідні керуючі сигнали та сигнали синхронізації для установки всіх регістрів системи у вихідний стан і завантаження констант фаз у селектори вибору по відповідних ланцюгах: з перших виходів вузла 5 на перші входи першого селектора вибору 17; із третіх виходів вузла 5 на другі входи другого селектора вибору 18; із других виходів вузла 5 на другі входи третього селектора вибору 19. При наявності декількох подібних чипів, об'єднаних у єдину систему, завантаження констант фаз у перший селектор вибору може бути також здійснене від HOST-машини або інших аналогічних чипів через інтерфейс завантаження 9 по ланцюзі: із третіх виходів інтерфейсу завантаження 9 на другі входи першого селектора вибору 17. Крім того, через інтерфейс завантаження 9 надходять команди початкового завантаження, які з перших виходів цього інтерфейсу направляються через відповідні перші входи банків пам'яті 1 та 2 у кеш-пам'яті команд 26 кожного банку пам'яті (Фіг.1, Фіг.2), а із других виходів цього інтерфейсу через треті входи вузла 16 кеш-пам'ятей команди початкового завантаження надходять у кеш-пам'ять команд 14 кожного ведучого процесора 13. Керування операціями завантаження в кеш-пам'ять здійснює контролер пам'яті 3, із третіх виходів якого на другі входи інтерфейсу завантаження 9 надходять сигнали керування, призначення та послідовність яких визначається керуючими сигналами, які надходять на його четверті входи із сьомих виходів вузла 5, сформованих у результаті обробки цим вузлом відповідного поля пакета. Далі, використовуючи програми початкового завантаження, відбувається завантаження робочих програм і необхідних даних у масиви пам'яті 22 (Фіг.2) банків пам'яті 1 та 2 або від зовнішньої пам'яті через інтерфейс зовнішньої пам'яті 4, або від пристроїв вводу-виводу, або інших аналогічних чипів через інтерфейс вводу-виводу 8 під керуванням контролера пам'яті 3. Для виконання завантаження в масиви пам'яті 22 використовуються в кожному конкретному випадку одна з описаних вище модифікацій операцій запису інформації, що надходить від пристроїв, які перебувають поза кристалом пам'яті.

Налаштування архітектури пропонованої системи пам'яті. Закладені в пропоновану систему пам'яті функціональні можливості дозволяють виконати кілька різновидів налаштувань архітектури та структури системи залежно від вимог, пропонованих до системи в цілому та типу розв'язуваного користувальницького завдання. Максимальна кількість елементів на кристалі визначається, насамперед, можливостями інтегральної технології (головним чином - розмірами одного елемента та розмірами кристала), у значній мірі складністю й регулярністю розташовуваних на кристалі схем, а також максимально допустимою потужністю, що розсіює кристал, що більшою мірою визначається кількістю одночасно працюючих логічних елементів, ніж елементів пам'яті. Тому при створенні системи пам'яті на одному кристалі знаходиться компроміс між максимальною кількістю елементів пам'яті та максимальною кількістю логічних елементів, які можуть бути розміщені разом з елементами пам'яті. Ці параметри, що є обмеженнями для налаштування архітектури, визначаються відразу при виготовленні мікросхеми й не підлягають зміні. Тому основні параметри архітектури, які підлягають налаштуванню, визначаються метою корисної моделі - розширення функціональних можливостей системи пам'яті, підвищення продуктивності й зменшення споживаної потужності кристала. До таких параметрів відносяться:  $s$  - кількість використовуваних проблемно-орієнтованих (спеціалізованих) ведучих процесорів, оптимальних по своїх параметрах для рішення різних класів завдань;  $m$  - кількість і розрядність  $R$  одночасно використовуваних (обраних) для кожного застосування банків пам'яті;  $q$  - кількість і розрядність  $r$  одночасно використовуваних підбанків пам'яті в складі обраних банків пам'яті; функції та властивості використовуваних комунікаційних засобів, що дозволяють здійснювати вибір необхідних ресурсів системи відповідно до карти розподілу ресурсів й установити взаємозв'язок між обраними та іншими компонентами системи пам'яті для передачі інформації з необхідних напрямків.

Розширення функціональних можливостей і підвищення продуктивності досягається шляхом застосування набору ведучих процесорів 13, кожний з яких орієнтований або спеціалізований на рішення певного класу завдань або фрагмента великого завдання й тому має екстремальні значення відповідних параметрів. Наприклад, один із процесорів може бути орієнтований на рішення завдань обробки графічних зображень, інший - на рішення системи лінійних алгебраїчних рівнянь, третій - на обробку векторних операцій і т.д. Тому для кожного додатка, для кожної паралельної гілки реалізації алгоритму, а також для відповідних фрагментів алгоритму кожної гілки у відповідні моменти часу вибирається найбільш придатний по параметрах тип ведучого процесора 13 з наявного на кристалі набору процесорів. При цьому щораз для кожного обраного процесора 13 виділяється конкретний набір ресурсів (у першу чергу - кількість використовуваних банків і підбанків пам'яті і їхніх параметрів) у суворій відповідності з картою розподілу ресурсів, приклад якої наведений на Фіг.5. Карта розподілу ресурсів, подібна наведеної на Фіг.5, створюється компілятором завдання на зовнішній, стосовно кристала, ЕОМ (HOST-машині). Інформація про цю карту передається у вузол 5 чипа пам'яті, де запам'ятовується на час реалізації застосування та піддається відповідному аналізу й обробці. Якщо ж дозволяє наявна площа кристала, то подібна карта розподілу ресурсів може бути створена спеціальними засобами вузла 5, використовуючи при цьому вихідну інформацію відповідних полів керуючого пакета, що надходить на перші входи вузла 5.

Відповідно до цього перший етап настроювання архітектури полягає в тому, щоб відповідно до карти розподілу ресурсів перед початком реалізації застосування вибрати необхідний тип ведучого процесора 13. Це досягається шляхом видачі на першому виході вузла 5 сигналу "Скидання", що надходить на відповідний вхід групи перших входів першого селектора вибору й далі на третій однойменний вхід регістра керування вибором 33 і на другий вхід лічильника фаз 30, встановлюючи всі розряди коду регістра 33 і лічильника фаз 30 у стан "нуль". Цей код з виходу лічильника фаз 30 надходить на входи дешифратора констант 31 (Фіг.3), на першому виході якого з'являється сигнал вибірки першого за часом видачі коду константи, розміщеної в пам'яті констант 32. Код константи, зчитаний з пам'яті констант і визначаючий початок роботи ведучих процесорів, надходить на перші входи однойменних розрядів регістра 33, де тимчасово запам'ятовується до появи наступного (другого) коду константи. Далі з виходів регістра 33 код константи надходить на відповідні перші входи провідних процесорів 13, дозволяючи роботу тільки тих процесорів, на перших входах яких є присутнім з виходу другого селектора 18 сигнал дозволу - логічна "одиниця".

Аналогічно відбувається другий етап настроювання вихідної архітектури системи пам'яті по підключенню до обраних ведучих процесорів 13 основних ресурсів системи, які будуть задіяні на початковому етапі реалізації алгоритму: банків пам'яті 1. Для вибору відповідно до карти розподілу ресурсів банків пам'яті із третього виходу вузла 5 на другий вхід другого селектора вибору 18 надходить сигнал "Скидання", установлюючи лічильник фаз 30 і регістр керування вибором 33 у стан "нуль" (Фіг.4). Точно також "нульовий" код (значення всіх розрядів коду лічильника фаз рівні "нулю") надходить як адреса на входи дешифратора констант 31 (Фіг.4), на першому виході якого з'являється сигнал вибірки першого за часом видачі коду константи, розміщеної в пам'яті констант 32. Код константи, зчитаний з пам'яті констант і визначаючий початок роботи обраних банків пам'яті, надходить на перші входи однойменних розрядів регістра 33, де тимчасово запам'ятовується до появи наступного (другого) коду константи. Далі з виходів регістра 33 (Фіг.4) код константи надходить на відповідні другі входи банків пам'яті 1 та 2 (Фіг.1) і далі на треті входи дешифраторів 23, дозволяючи роботу на першому кроці реалізації алгоритму тільки тих банків пам'яті, у яких на третій вході дешифраторів 23 є присутнім з виходу другого селектора сигнал дозволу - логічна "одиниця" (Фіг.2).

Третій етап настроювання вихідної архітектури полягає у виборі необхідної кількості підбанків пам'яті 20 для кожного обраного банку й відповідних зв'язків, що забезпечують ефективну взаємодію на момент запуску системи між компонентами в межах підбанку, банку пам'яті, між банками пам'яті та пристроями, які підключені до кристала пам'яті. Цей процес здійснюється за допомогою третього селектора вибору 19 (Фіг.1), на другий вхід якого на етапі вихідного настроювання надходить із другого виходу вузла 5 сигнал "Скидання". При цьому третій селектор вибору працює аналогічно другому селектору вибору 18, виставляючи на виході регістра 33 (Фіг.4) код керування вибором підбанків пам'яті, кожен вихід якого з'єднаний з відповідним третім входом банків пам'яті та через нього - з п'ятим входом (дозволу) регістра 27 (Фіг.2) і четвертим входом (дозволу) процесорного ядра 25. При цьому відповідно до таблиці розподілу ресурсів той підбанк пам'яті буде включений у роботу на початковому етапі функціонування системи, на п'ятому вході регістра 27 і на четвертому вході процесорного ядра 25 якого встановлений сигнал дозволу - логічна "одиниця" з відповідного виходу третього селектора 19.

Аналогічним способом відбувається перебудова архітектури (вибір іншого ведучого процесора, інших банків і підбанків пам'яті) системи в процесі її функціонування. Відмінність складається лише в тому, що на відповідні входи лічильників фаз 30 першого 17, другого 18 і третього 19 селекторів вибору (Фіг.3, Фіг.4) надходить відповідно з першого, третього та другого виходів вузла 5 послідовність ознак фаз ВР (верхнього рівня) для вибору ведучих процесорів, послідовність фаз СР (середнього рівня) для вибору банків пам'яті та послідовність фаз НР (нижнього рівня) для вибору підбанків пам'яті. Ці послідовності в кожному селекторі вибору надходять на входи відповідних лічильників фаз 30, формуючи з виходів всіх розрядів лічильника код, що є кодом адреси для вибору з пам'яті констант фаз 32 відповідного коду константи, що була занесена в пам'ять в режимі завантаження. Логічна "одиниця" на відповідних розрядних позиціях коду константи є сигналом дозволу для функціонування в певний момент часу обраних об'єктів у суворій відповідності з таблицею розподілу ресурсів (ведучих процесорів, банків і підбанків пам'яті).

Настроювання зв'язків між компонентами системи відбувається в такий спосіб. Для реалізації описаних вище операцій "Читання", "Запис", "Пересилання" і їхніх модифікацій у пропонованій системі пам'яті застосовані комутаційні схеми та вузли, які умовно можна розділити на два комутаційні поля. Ланцюги комутації першого комутаційного поля розосереджені по входах і виходах схем пам'яті системи (реєстрових схем, кеш-пам'яті) і забезпечують комутацію інформації із двох напрямків. Перший напрямок - це комутація інформації між зовнішнім стосовно банків пам'яті регістром рядка 11 і регістрами підбанків 27 всіх банків пам'яті з використанням вихідних шин, підключених до других виходів кожного банку пам'яті та через другі входи схеми 12 до третіх входів регістра рядка 11, і вхідних шин, що з'єднують четверті входи цього регістра із сьомими входами банків пам'яті. При цьому кожен регістр 27 містить вхідну логіку 28 і вихідну логіку 29, які управляються сигналом "Тип А" з п'ятих виходів контролера пам'яті 3 (Фіг.1). Другий напрямок - це двоспрямована комутація між регістром рядка 11 (через селектор читання 6 і декодер запису 7) і вузлом 16 кеш-пам'яті (Фіг.1). При цьому використовуються к-

розрядні шини, підключені до третіх виходів селектора читання й до других відповідних входів вузла кешей 16, а також k-розрядні шини, що з'єднують другі виходи вузла кеш-пам'яті 16 з відповідними четвертими входами декодера запису 7. Керування комутацією при реалізації режимів читання і їхніх модифікацій здійснюється за допомогою селектора читання 6, що за інформацією, отриманою через його четвертий вхід від третього селектора вибору, а також за значенням коду адреси стовпця на третьому вході селектора читання, формує стробіруючий сигнал, що виділяє на довгому N-розрядному рядку, розміщеному на регістрі 11, k-розрядну область потрібного слова ( $k \leq N$ ) для зчитування та передачі виділеного слова по відповідних шинах у вузол кеш-пам'яті 16, або за межі чипа через інтерфейс вводу-виводу 8 або через паралельний інтерфейс до зовнішньої пам'яті 4.

Керування комутацією при реалізації режимів запису і їхніх модифікацій здійснюється за допомогою декодера запису 7, що за інформацією, отриманою через його п'ятий вхід від третього селектора вибору, а також за значенням коду адреси стовпця на третьому його вході, формує стробіруючий сигнал, що виділяє на довгому N-розрядному рядку, розміщеному на регістрі 11, k-розрядну область ( $k \leq N$ ) для запису k-розрядного слова, отриманого із других виходів інтерфейсу 4 або із других виходів інтерфейсу 8 у виділену область на регістрі 11, з виходів якого це слово може бути передане в масиви 22 банків пам'яті або у вузол кеш-пам'яті 16.

Ланцюги комутації другого комутаційного поля представлені в кожному банку пам'яті 20 комутатором 21, регістром 27 із вхідною 28 і вихідною 29 логікою й забезпечують комутацію інформації усередині кожного банку пам'яті також по двох напрямках. Перший напрямок - це комутація інформації між вихідною логікою 29 кожного регістра 27 будь-якого підбанку пам'яті 20 і будь-яким процесорним ядром 25 цього банку пам'яті за допомогою комутатора 21. Керування такою комутацією здійснюється за допомогою сигналу "Тип А", що надходить на треті входи вихідної логіки 29, а також за допомогою керуючого сигналу "Комутація", що надходить на третій вхід комутатора через восьмі входи кожного банку пам'яті із четвертих виходів вузла інтерфейсів та обробки керуючого пакета 5.

Другий напрямок - це комутація інформації з виходу процесорного ядра 25 підбанку пам'яті через вхідну логіку 28 і вихідну логіку 29 цього підбанку на відповідні входи буферної схеми 24 і далі в масив пам'яті 22 цього ж підбанку. При цьому керування такою комутацією здійснюється (Фіг.2) за допомогою сигналу "Тип А", що надходить на відповідні треті входи вхідної логіки 28 і вихідної логіки 29, а також на другий вхід комутатора 21 через п'яті входи банку пам'яті з п'ятих виходів контролера пам'яті 3 (Фіг.1).

Крім того, архітектура системи пам'яті може бути налаштована таким чином, що при реалізації масових операцій усередині кристала схема виявлення й виправлення помилок 12 разом з банком пам'яті контрольних кодів 2 може бути включена та включена в роботу лише при видачі результатів за межі банків пам'яті 1. Керування такою комутацією здійснюється за допомогою керуючого сигналу, що надходить на третій вхід схеми 12 з дев'ятого виходу вузла інтерфейсів й обробки керуючого пакета 5, а також за допомогою сигналу, що надходить на другий вхід банку пам'яті 2 з відповідного виходу групи перших виходів другого селектора вибору 18.

Обробка інформації в режимі СО. У цьому режимі роботи запропонованої системи пам'яті відбувається розпаралелювання інформації на декількох рівнях:

- на рівні завдань (великих фрагментів алгоритму) по паралельних гілках з відповідним розподілом для кожного завдання ведучих процесорів і необхідних до них ресурсів відповідно до карти розподілу ресурсів, спрощений приклад якої наведений на Фіг.5;
- на рівні послідовних ланцюжків зв'язаних макрооперацій усередині однієї паралельної гілки;
- на рівні команд реалізації макрооперацій.

Перший рівень розглянутий досить докладно при описі складу й принципів функціонування системи пам'яті. При розгляді другого рівня розпаралелювання в режимі СО будемо використовувати макрооперації "Читання/СО", "Запис/СО", "Пересилання/СО", описи їхніх модифікацій, які наведені вище, а також схему системи пам'яті та схему банку пам'яті, наведені на Фіг.1 і Фіг.2 відповідно. При цьому будемо розглядати організацію процесу обробки інформації для однієї паралельної гілки при участі одного ведучого процесора 13 та одного банку пам'яті 1, а також будемо мати на увазі, що результат виконання процесорним ядром 25 будь-якої операції розміщується на регістрі 27 відповідного підбанку пам'яті 20.

Реалізація процесорним ядром 25 будь-якої арифметичної або логічної операції в позначених вище макроопераціях можна представити у вигляді трьох послідовно зв'язаних етапів: підготовка операндів - власне операція над цими операндами - видача результатів. У такій інтерпретації розглянемо кілька прикладів.

Якщо за допомогою процесорного ядра (ПЯ) 25 необхідно виконати арифметичну або логічну операцію над даними, які розміщені в масиві пам'яті 22 банку пам'яті 20, а отриманий результат операції також необхідно розмістити в масиві пам'яті 22, то ланцюжок макрооперацій у наведених вище позначеннях буде мати вигляд:

- 1). "Читання 2/СО" → "Пересилання 3/СО" → (операція ПЯ) → "Запис 3/СО".

Якщо отриманий результат виконання цієї операції процесорним ядром повинен бути записаний у регістр рядка 11, то:

- 2). "Читання 2/СО" → "Пересилання 3/СО" → (операція ПЯ) → "Пересилання 4/СО" → "Пересилання 1/СО".

Якщо отриманий результат виконання цієї ж операції процесорним ядром повинен бути записаний у кеш-пам'ять даних 15, то:

- 3). "Читання 2/СО" → "Пересилання 3/СО" → (операція ПЯ) → "Пересилання 4/СО" → "Пересилання 1/СО" → "Запис 2/СО".

Якщо виконується процесорним ядром 25 арифметична або логічна операція над даними, які є результатами виконання операцій ведучого процесора 13, при цьому отриманий результат операції необхідно розмістити в кеш-пам'ять даних 15 вузла 16, то ланцюжок макрооперацій буде мати вигляд:

- 4). "Запис 1/СО" → "Читання 1/СО" → "Пересилання 2/СО" → (операція ПЯ) → "Пересилання 4/СО" → "Пересилання 1/СО" → "Запис 2/СО".

При використанні даних, що надходять із пристроїв поза чипом і записом результату операції в масив пам'яті 22 ланцюжок макрооперацій буде мати вигляд:

- 5). "Читання 3/СО" → "Пересилання 2/СО" → (операція ПЯ) → "Запис 3/СО".

Та ж операція з видачею результатів у кеш-даних 15 вузла 16 може бути представлена у вигляді:



6). "Читання 3/CO" → "Пересилання 2/CO" → (операція ПЯ) → "Пересилання 4/CO" → "Пересилання 1/CO" → "Запис 2/CO".

Аналогічно можна представити інші ланцюжки макрооперацій при різних вихідних даних і різних напрямках видачі отриманого результату.

Приклад можливої реалізації зазначених шести ланцюжків макрооперацій на шкалі часу для одного банку пам'яті показано на Фіг.6, при цьому прийняті наступні позначення згідно Фіг.1 - Фіг.3: КЕШ15 - кеш-пам'ять даних 15 вузла кеш 16; РrP11 - регістр рядка 11; M22 - масив пам'яті 22 підбанку пам'яті 20; Рr27 - регістр 27 підбанку пам'яті; ІП - інтерфейс до зовнішньої пам'яті 4; Ів/в8 - інтерфейс вводу-виводу 8; ВР13 - ведучий процесор 13; ПЯ25 - процесорне ядро 25; К21 - комутатор 21.

Прийнято, що пересилання регістр-регістр виконується за два такти, операції кеш-пам'ять 15 - регістр або регістр - кеш-пам'ять 15 виконується за три такти, операції масив пам'яті 22 - регістр або регістр - масив пам'яті 22 виконується за чотири такти, затримка комутатора 21 - два такти, а середня тривалість виконання операції процесорним ядром 25 - п'ять тактів. Навіть при такій наближеній розкладці часових параметрів можна показати, що при реалізації будь-якої паралельної гілки А, В, С або D (Фіг.5) на тому самому обладнанні тільки одного банку пам'яті й тільки для одного процесорного ядра можливо сполучити в часі виконання, принаймні, двох різних ланцюжків. На Фіг.7 показаний один з можливих варіантів суміщення першого і п'ятого, а також третього і шостого позначених вище ланцюжків макрооперацій.

Що стосується розпаралелювання операцій на третьому рівні - на рівні команд, то можна відзначити, що операції вводу інформації на регістр 11 через інтерфейс зовнішньої пам'яті 4 або через інтерфейс вводу-виводу 8 точно також, як й операції виводу інформації через ці ж інтерфейси, можуть бути сполучені в пропонованій системі пам'яті з різними операціями обробки усередині кристала. Процедури такого суміщення очевидні й тому докладно не розглядаються.

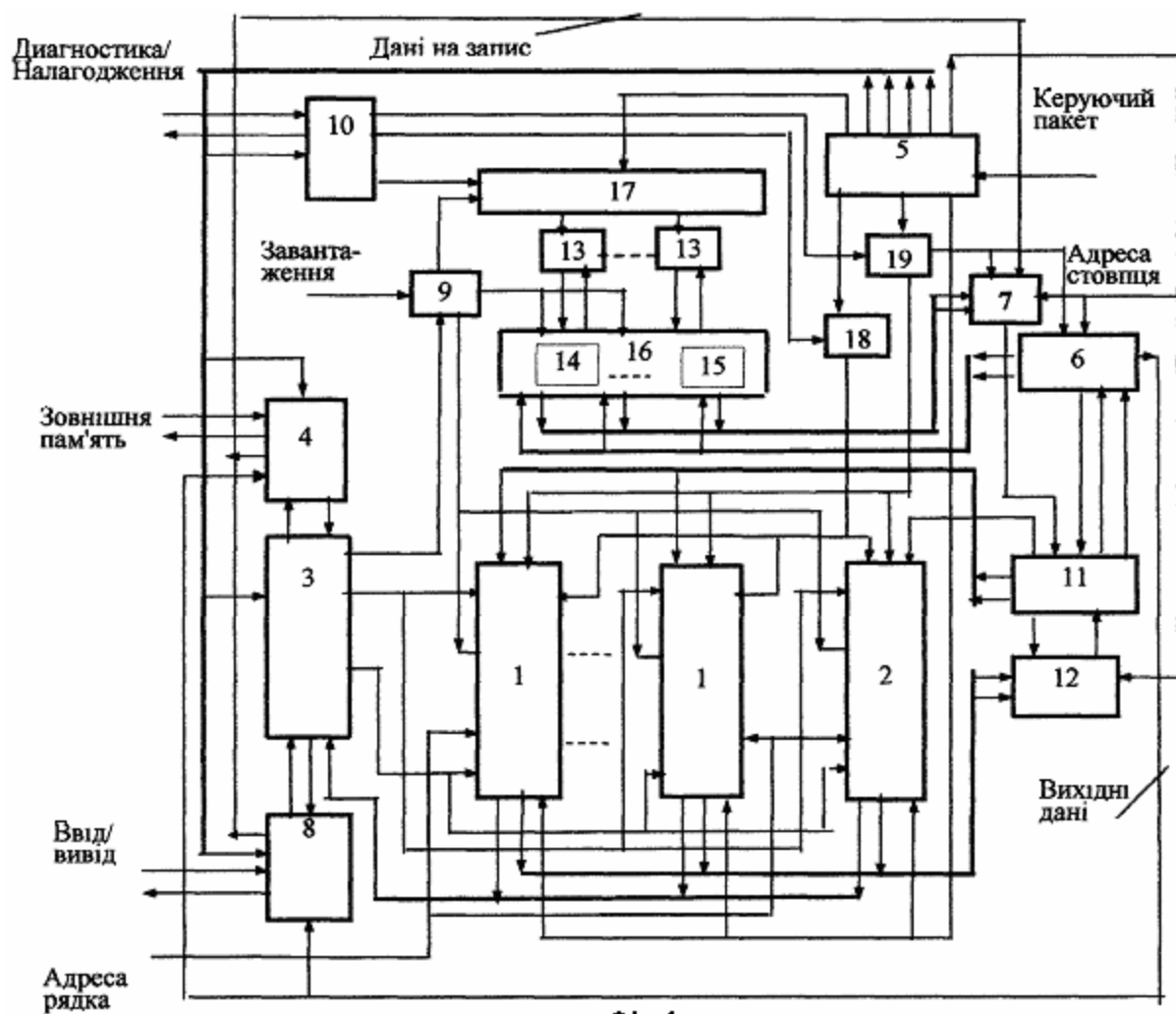
Крім того, введення в пропоновану систему пам'яті набору ведучих процесорів 13, кожний зі своєю кеш-пам'яттю даних 15, а також у кожен підбанк пам'яті регістра 27 із вхідною логікою 28 і вихідною логікою 29, дозволяє організувати при виконанні операцій в "режимі СО" багаторівневий конвеєр. При цьому підтримку першого рівня забезпечує ведучий процесор з кеш-пам'яттю команд і кеш-пам'яттю даних, другого рівня - вузол кеш-пам'яті 16 і регістр рядка 11 разом із селектором читання 6 і декодером запису 7, третього рівня - регістр рядка 11 і регістри 27 разом із вхідною логікою 28 і вихідною логікою 29 у кожному підбанку пам'яті 20, підтримку четвертого рівня - регістри рядка 27, комутатор 21 і вхідні регістри процесорних ядер 25 та, нарешті, по крайній мірі, два рівні забезпечені структурою процесорного ядра 25.

Таким чином, наявність набору ведучих процесорів, орієнтованих (і навіть спеціалізованих) на різні класи завдань, дозволяє істотно розширити функціональні можливості системи за рахунок сукупності функціональних можливостей всіх ведучих процесорів і підключення до них необхідних ресурсів за допомогою введених засобів комутації й вибору. Наявна в пропонованій системі можливість за рахунок введення першого селектора вибору з відповідними зв'язками дозволяє при необхідності вибрати оптимальний тип ведучого процесора як перед початком роботи системи, так й у динаміці, що забезпечує на конкретних застосуваннях істотне підвищення (у порівнянні із прототипом) продуктивності системи в цілому.

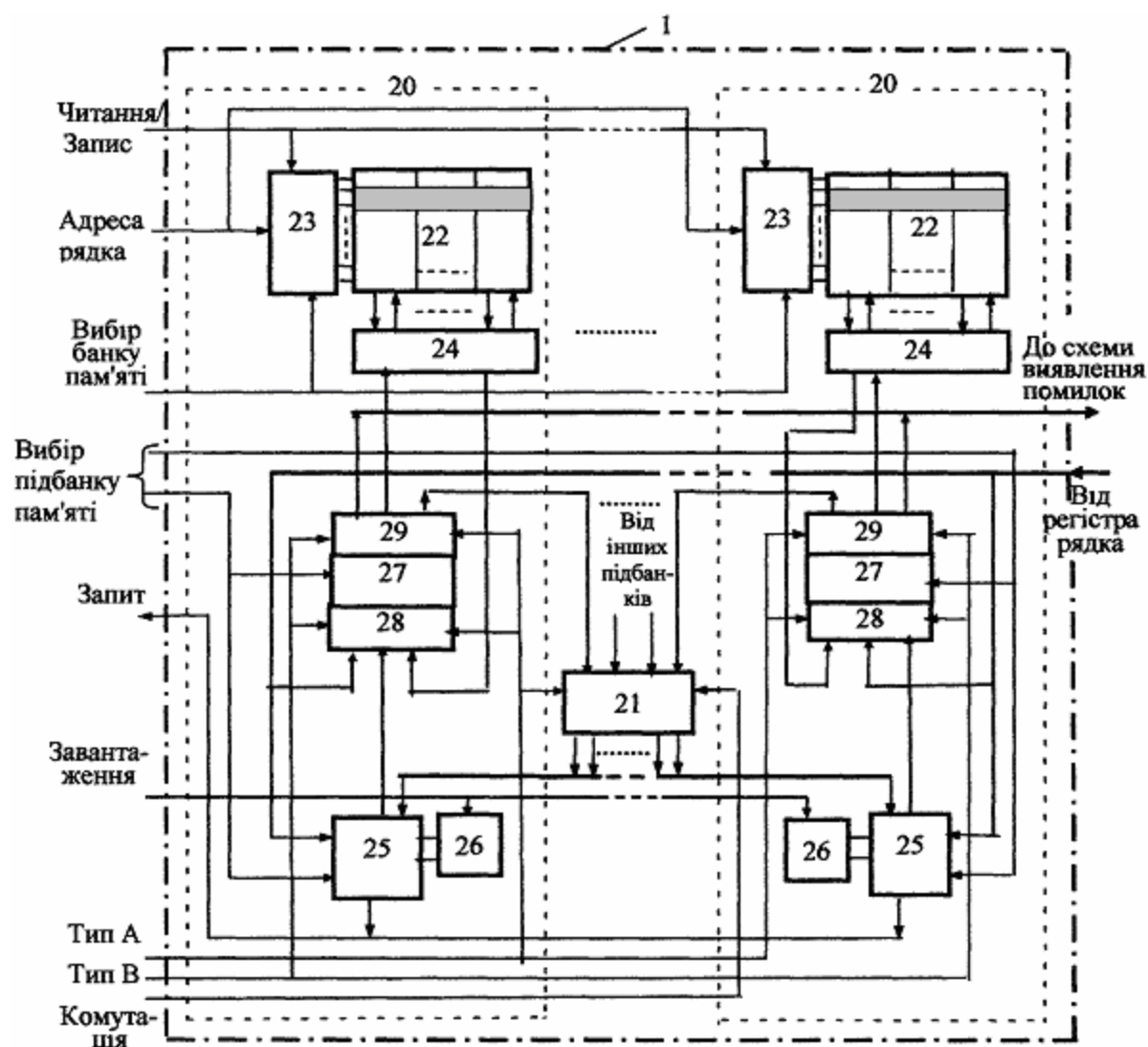
Реалізована в пропонованій системі пам'яті можливість вибирати в потрібні моменти часу на певних етапах реалізації застосування необхідну кількість ресурсів (банків і підбанків пам'яті разом із процесорними ядрами) шляхом використання введених у систему другого та третього селекторів вибору з відповідними зв'язками, дозволяє виключити при виконанні різних операцій необхідність працювати з повним N-розрядним рядком (замість k-розрядного слова,  $k \ll N$ ), тобто виключити "холосту" роботу великої кількості процесорних ядер у банках пам'яті та масове перемикання логічних елементів й елементів пам'яті. Це дозволяє знизити споживану кристалом потужність і підвищити перешкодозахищеність системи, а також звільнені ресурси використати для додаткового розпаралелювання процесу, і, отже, - підвищити продуктивність.

Сукупність зазначених вище введених у систему пам'яті засобів і зв'язків забезпечує можливість розпаралелювання процесу обробки інформації на трьох рівнях: на верхньому рівні - розпаралелювання на паралельні гілки завдань або великих фрагментів завдань із виділенням необхідних ресурсів (Фіг.5), на середньому рівні - розпаралелювання кожної паралельної гілки шляхом суміщення в часі реалізації великих ланцюжків макрооперацій (Фіг.6, Фіг.7) і на нижньому рівні (рівні команд) - розпаралелювання процедур та операцій по обробці й передачі інформації за межі кристала. Усе це забезпечує можливість істотного (у порівнянні із прототипом) підвищення продуктивності. Крім того, наявна в системі можливість відключати в певні моменти часу при масових операціях усередині кристала схему контролю та виправлення помилок також істотно підвищує продуктивність системи, оскільки із загального циклу виконання масових операцій виключається значний час затримки, затрачуваний засобами контролю й виправлення помилок.

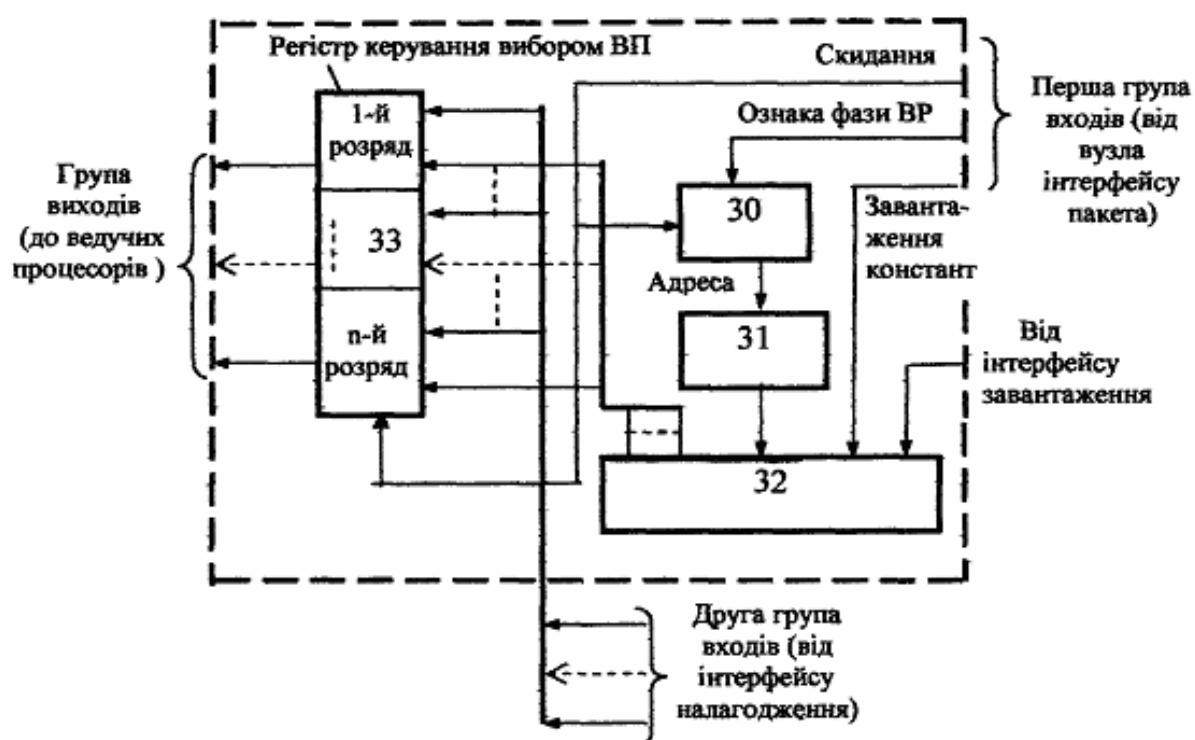
Таким чином, розширення в порівнянні із прототипом функціональних можливостей системи пам'яті й підвищення продуктивності для широкого класу завдань (і навіть для різних класів завдань) дозволяє стверджувати про широку область застосування пропонованого рішення, що, у свою чергу, може привести до масового виробництва кристала системи пам'яті та істотного зниження його вартості. Поява та широке використання мікросхем зі структурою, що перебудовується (типу ПЛИС) може істотно скоротити строки та вартість розробки на базі пропонованого чипа системи пам'яті.



Фиг. 1



Фіг. 2



Фіг. 3



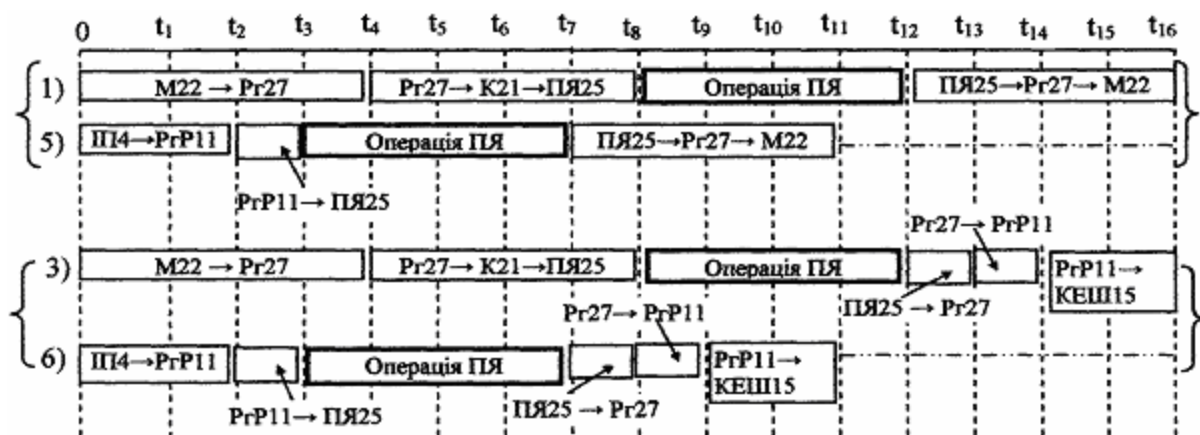
Фіг. 4

Шкала часу (умовні одиниці)	Розподіл ведучих процесорів ВП <sub>s</sub> (s = 1,...,4) і банків пам'яті m <sub>i</sub> (i = 1,..., 4) у паралельних гілках А, В, С, Д алгоритма											
	А		В		С		Д		Ознаки фаз ВР		Ознаки фаз СР	
	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>2</sub>	m <sub>2</sub>	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>4</sub>	m <sub>4</sub>	Такт	Код	Такт	Код
	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>2</sub>	m <sub>2</sub>	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>4</sub>	m <sub>4</sub>	T <sub>1</sub>	1111	t <sub>1</sub>	1111
1	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>2</sub>	m <sub>2</sub>	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>4</sub>	m <sub>4</sub>				
2	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>2</sub>	m <sub>2</sub>	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>4</sub>	m <sub>4</sub>			t <sub>2</sub>	
3	ВП <sub>1</sub>	m <sub>2</sub>	ВП <sub>2</sub>	—	ВП <sub>3</sub>	m <sub>1</sub>	ВП <sub>4</sub>	m <sub>4</sub>			t <sub>3</sub>	1101
4	ВП <sub>1</sub>	m <sub>2</sub>	ВП <sub>2</sub>	m <sub>3</sub>	ВП <sub>3</sub>	m <sub>1</sub>	ВП <sub>4</sub>	m <sub>4</sub>				1111
5	ВП <sub>1</sub>	m <sub>2</sub>	ВП <sub>2</sub>	m <sub>3</sub>	ВП <sub>3</sub>	m <sub>1</sub>	ВП <sub>4</sub>	m <sub>4</sub>	T <sub>2</sub>		t <sub>4</sub>	
6	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>2</sub>	m <sub>4</sub>	ВП <sub>3</sub>	m <sub>3</sub>	—	—		1110		1011
7	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>2</sub>	m <sub>4</sub>	ВП <sub>3</sub>	m <sub>3</sub>	—	—	T <sub>3</sub>		t <sub>5</sub>	
8	—	—	ВП <sub>2</sub>	m <sub>4</sub>	ВП <sub>3</sub>	m <sub>3</sub>	—	—	T <sub>4</sub>	0110	t <sub>6</sub>	0011
9	ВП <sub>2</sub>	m <sub>2</sub>	—	—	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>4</sub>	m <sub>4</sub>		0111		0111
10	ВП <sub>2</sub>	m <sub>2</sub>	—	—	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>4</sub>	m <sub>4</sub>	T <sub>5</sub>		t <sub>7</sub>	
11	ВП <sub>2</sub>	m <sub>2</sub>	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>4</sub>	m <sub>4</sub>		1111		1111
12	ВП <sub>2</sub>	m <sub>2</sub>	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>4</sub>	m <sub>4</sub>				
13	ВП <sub>2</sub>	m <sub>2</sub>	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>4</sub>	m <sub>4</sub>				
14	ВП <sub>2</sub>	m <sub>2</sub>	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>1</sub>	m <sub>1</sub>	ВП <sub>4</sub>	m <sub>4</sub>	T <sub>6</sub>			
15	ВП <sub>2</sub>	m <sub>2</sub>	—	—	—	—	ВП <sub>4</sub>	m <sub>4</sub>	T <sub>7</sub>	0101	t <sub>8</sub>	0111
16	ВП <sub>1</sub>	m <sub>1</sub>	—	—	ВП <sub>3</sub>	m <sub>3</sub>	ВП <sub>4</sub>	m <sub>4</sub>		1011	t <sub>9</sub>	1011

Фіг. 5

Найменування макрооперації	Зміст макрооперації	Тривалість у тактах для номера ланцюжка макрооперацій					
		1	2	3	4	5	6
"Читання 1/CO"	КЕШ15 → PrP11				$t_3 - t_5$		
"Читання 2/CO"	M22 → Pr27	$t_1 - t_4$	$t_1 - t_4$	$t_1 - t_4$			
"Читання 3/CO"	П4 → PrP11 Ів/в8 → PrP11					$t_1 - t_2$	$t_1 - t_2$
"Запис 1/CO"	ВП13 → КЕШ15				$t_1 - t_3$		
"Запис 2/CO"	PrP11 → КЕШ15			$t_{14} - t_{16}$	$t_{12} - t_{14}$		$t_9 - t_{11}$
"Запис 3/CO"	ПЯ25 → Pr27 → M22	$t_{12} - t_{13} - t_{16}$				$t_7 - t_8 - t_{11}$	
"Пересилання 1/CO"	Pr27 → PrP11		$t_{13} - t_{14}$	$t_{13} - t_{14}$	$t_{11} - t_{12}$		$t_8 - t_9$
"Пересилання 2/CO"	PrP11 → ПЯ25 PrP11 → Pr27				$t_5 - t_6$	$t_2 - t_3$	$t_2 - t_3$
"Пересилання 3/CO"	Pr27 → K21 → ПЯ25	$t_4 - t_6 - t_8$	$t_4 - t_6 - t_8$	$t_4 - t_6 - t_8$			
"Пересилання 4/CO"	ПЯ25 → Pr27		$t_{12} - t_{13}$	$t_{12} - t_{13}$	$t_{10} - t_{11}$		$t_7 - t_8$
Операція ПЯ	Арифметична або логічна операція	$t_8 - t_{12}$	$t_8 - t_{12}$	$t_8 - t_{12}$	$t_6 - t_{10}$	$t_3 - t_7$	$t_3 - t_7$

Фіг. 6



Фіг. 7