

## Висновки

1. Запропоновано модель архітектури програмованого комп'ютерного пристрою як множину параметрів: багаторівнева множина його компонентів та їх функцій, системи зв'язків між компонентами на всіх рівнях, множина форматів команд, множина форматів адрес, множина параметрів даних, склад системи команд, множина способів адресації пам'яті.

2. Формалізовано опис вищеназваних складових та показаний їх взаємозв'язок через вплив параметрів архітектури програмованого комп'ютерного пристрою на його характеристики.

3. Запропоновані модель та формалізований опис дають можливість визначення взаємозв'язків між компонентами архітектури ПКП та визначення їх впливу на його характеристики, що сьогодні є особливо актуальним у зв'язку з необхідністю подальшого підвищення ефективності комп'ютерних засобів.

1. Amdahl, G. M. [1967]. "Validity of the single processor approach to achieving large scale computing capabilities," *Proc. AFIPS 1967 Spring Joint Computer Conf.* 30 (April), Atlantic City, N.J., 483–485. 2. Мельник А.О. Програмовані процесори обробки сигналів. – Львів: „Львівська політехніка”, 2000. – 57 с. 3. Мельник А.О. Архітектура комп'ютера. – Луцьк: Волинська обласна друкарня, 2008. – 470 с. 4. Burks, A. W., H. H. Goldstine, and J. von Neumann [1946]. "Preliminary discussion of the logical design of an electronic computing instrument," *Report to the U.S. Army*. 5. Tanenbaum, A. *Structured Computer Organization*, 4<sup>th</sup> ed. Upper Saddle River, NJ: Prentice Hall, 1999.

УДК 004.31, 004.056.55, 003.26

А.О. Мельник, Д.Х. Аль Равашдех

Національний університет "Львівська політехніка",  
кафедра електронно-обчислювальних машин

## ПОРІВНЯЛЬНИЙ АНАЛІЗ ТИПІВ ПАМ'ЯТІ КОМП'ЮТЕРА

© Мельник А.О., Аль Равашдех Д.Х., 2008

**Сформовано вимоги до пам'яті сучасного комп'ютера та критерії її ефективності під час використання комп'ютера для виконання інтенсивних обчислень над масивами даних. Виконано аналіз та порівняння пам'яті сучасних комп'ютерів за вибраними критеріями. Визначено та обґрунтовано переваги та вузькі місця і недоліки пам'яті з довільним, послідовним, асоціативним та програмованим доступом.**

**The requirements for computer memory and the criteria of its efficiency for computer using to intensive computing at data arrays are defined. The analysis and comparison of computer memory with defined criteria have been carried out. The advantages and weakness of random access memory, serial access memory, associative access memory and programmable access memory have been found out.**

**Вступ.** Пам'ять є одним з базових вузлів комп'ютера. Вона необхідна для зберігання інформації (команд та даних) та забезпечення обміну нею з пристроями комп'ютера [1–4]. З розвитком комп'ютерів істотні зміни відбулися і в будові пам'яті, значно покращилися їх технічні характеристики [5]. До того ж виникає потреба в створенні пам'яті з новими властивостями, якими не володіють наявні типи пам'яті. Зокрема, під час виконання інтенсивних обчислень над масивами даних, як це необхідно, наприклад, під час розв'язання мультимедійних та телекомунікаційних

задач, потрібно забезпечувати зберігання масивів даних, що надходять з багатьох каналів, одночасно з зчитуванням раніше прийнятих масивів даних для опрацювання в багатоблокових операційних пристроях, тобто паралельно виконувати операції реорганізації масивів даних та впорядкування даних у масивах [6]. Виконання цих функцій на існуючих типах пам'яті є доволі складною, а часто і невирішуваною з прийнятними характеристиками задачею, що зумовлено потенційними обмеженнями існуючих типів пам'яті. Ці обмеження істотно ускладнюють організацію роботи комп'ютера та спричиняють сповільнення його роботи. Постає задача підвищення ефективності пам'яті у разі її застосування для вищезазначених задач роботи з масивами даних.

**Огляд літератури.** Залежно від призначення і особливостей реалізації пристроїв комп'ютерної пам'яті, по-різному розглядають і питання їхньої класифікації. У комп'ютері використовують різні типи пам'яті, які, залежно від способу доступу до інформації, можуть бути класифіковані так [1, 5, 7]:

- Пам'ять з довільним доступом. До довільної комірки цієї пам'яті в кожному такті може бути записане дане, або з довільної комірки цієї пам'яті в кожному такті може бути зчитане дане.
- Пам'ять з асоціативним доступом. У такій пам'яті дані шукають за їх змістом або за деякою їх ознакою.
- Пам'ять з послідовним доступом, до якої дані записуються та з якої дані зчитуються послідовно одне за одним.

У роботах [8–12] запропоновано принципи та варіанти побудови нового типу пам'яті, яку далі будемо називати пам'яттю з програмованим доступом. Як і пам'ять з послідовним доступом, пам'ять з програмованим доступом орієнтована на роботу з масивами даних. У цій пам'яті забезпечується доступ до даних у програмно встановленому порядку, тобто код, який надходить у пам'ять разом з даним, або під час його зчитування, вказує місце даного у вихідному масиві.

Кожний тип пам'яті має свої особливості, переваги та недоліки, які визначають місце використання відповідного типу пам'яті в комп'ютері. Проаналізуємо принципи організації роботи відомих типів пам'яті, оцінимо їх переваги та недоліки з позиції задоволення ними вимог вищезазначених критеріїв ефективності.

**Постановка задачі.** Основним обмежувальним фактором пам'яті є спосіб доступу до даних, який в ній реалізовано [1]. Адже описана область використання пам'яті вимагає одночасного паралельного доступу до даних у багатьох комірках пам'яті. Чи забезпечується такий доступ і якою мірою в сучасних типах пам'яті? Яка організація пам'яті є найефективнішою для роботи з масивами даних і за якими критеріями можна порівняти різні типи пам'яті? Пошуку відповідей на ці питання і присвячена ця робота, в якій розглядаються критерії ефективності пам'яті в описаних вище застосуваннях та порівнюються існуючі типи пам'яті за цими критеріями.

### **1. Вимоги до пам'яті та критерії її ефективності під час використання комп'ютера для виконання інтенсивних обчислень над масивами даних**

Як ми вже відзначали, у разі використання комп'ютера для виконання інтенсивних обчислень над масивами даних є потреба в створенні пам'яті з новими властивостями, якими не завжди володіють існуючі типи пам'яті. Розглянемо вимоги до пам'яті сучасного комп'ютера детальніше.

Сьогоднішні комп'ютери є паралельними. Тому, насамперед, пам'ять має бути паралельною та забезпечувати одночасний безконфліктний доступ до даних з багатьох каналів. Лише наявність паралельної пам'яті відкриває шлях до створення принципово нових комп'ютерів. Отже, звідси постали два перші критерії ефективності пам'яті:

- 1) пам'ять повинна бути багатоканальною (багатопортовою);
- 2) пам'ять повинна забезпечувати можливість одночасного безконфліктного доступу до даних з багатьох портів.

Крім того, під час виконання інтенсивних обчислень над масивами даних потрібно забезпечувати зберігання масивів даних, що надходять, одночасно з зчитуванням раніше прийнятих масивів, виконувати операції реорганізації масивів та впорядкування даних у масивах. Аналіз цих вимог показує, що потрібно ввести ще декілька критеріїв ефективності пам'яті:

3) пам'ять повинна забезпечувати одночасний запис даних, які надходять на її входи, та зчитування на виходи раніше записаних даних. При цьому тут можна говорити про два випадки – одночасне записування та зчитування як масивів даних, так і одиничних даних;

4) пам'ять повинна забезпечувати можливість виконання операцій реорганізації масивів та впорядкування даних у масивах.

Для прискорення взаємодії пристроїв комп'ютера з пам'яттю та скорочення затрат обладнання на елементи доступу до даних, що особливо актуально для пам'яті великої ємності, зі збільшенням якої затрати на елементи доступу зростають, а її швидкодія знижується, є також питання зменшення кількості звернень за даними, передусім, виключення необхідності двократного звернення під час записування і зчитування числа, що, крім збільшення часу пошуку даного, вимагає потребу зберігання інформації про місцезнаходження даного. Необхідно переглянути доцільність забезпечення доступу до кожної комірки з входу та виходу пам'яті або з об'єднаного входу/виходу пам'яті, що вимагає великих затрат на елементи доступу та сповільнює роботу пам'яті. Особливо це питання є актуальним для пам'яті, яка задовольняє перший критерій, тобто багатоканальну пам'ять, коли необхідно забезпечувати одночасний доступ з кожного каналу до кожної комірки пам'яті, що є проблематичною задачею, особливо при великих ємностях пам'яті. Отже, з'явилися ще три критерії ефективності пам'яті:

5) пам'ять не повинна вимагати звернення за даними як під час їх записування, так і зчитування;

6) організація пам'яті не повинна передбачати необхідність звернення до комірки пам'яті під час записування і зчитування числа;

7) організація пам'яті не повинна передбачати потребу зберігання інформації про місце перебування цього в пам'яті.

## 2. Задоволення існуючими типами пам'яті вимог критеріїв ефективності

**Пам'ять з довільним доступом.** Найчастіше в комп'ютері використовується пам'ять з довільним доступом (ПДД), або адресна пам'ять. Цю пам'ять поділяють на два типи: оперативний запам'ятовуючий пристрій (ОЗП), англійський термін Random Access Memory (RAM), та постійний запам'ятовуючий пристрій (ПЗП), англійський термін Read Only Memory (ROM) [1-5].

Пам'ять з довільним доступом складається з комірок, кожна з яких зберігає одиницю інформації, яка називається словом. Слова складаються із бітів зі значеннями 0 або 1. В слові є  $n$  бітів, де  $n$  – довжина слова. Кожен біт має свій номер. Нумерація бітів в слові здійснюється справа наліво, або зліва направо. Зазвичай слово має довжину  $n = 2^k$ , де  $k = 0, 1, 2, \dots$  біт, наприклад 8 ( $k = 3$ ), 32 ( $k = 5$ ) і т.д.

Комірки пам'яті нумеруються, тобто кожна з них має свій номер, або адресу. Ту саму адресу має і слово, яке зберігається в цій комірці. Місце розміщення слова в пам'яті називається адресою слова. Якщо пам'ять може зберігати  $M$  слів розрядністю  $n$  кожне, то в якості адреси використовуються числа від 0 до  $M-1$ .  $M$  адрес є адресним полем (простором) цього комп'ютера. Використовуючи двійкове кодування, необхідно  $m$  бітів для представлення всіх адрес, де  $m = \lceil \log_2 M \rceil$ . Значення в дужках означає більше ціле. Зазвичай пам'ять комп'ютера будують так, щоб  $M$  було кратним ступеню двійки, що дозволяє ефективніше використовувати розрядну сітку адреси, а також спрощує обробку адрес. Існує поняття "організація пам'яті", яке вказує на розрядність комірок і їх кількість в пам'яті, тобто  $n \cdot 2^m$ . Це значення вказує також ємність пам'яті в бітах. Пам'ять з довільним доступом виконує дві основні операції: вибірку Fetch (або зчитування Read) і запам'ятовування Store (або запис Write). Робота цієї пам'яті організована так. У режимі записування на адресний вхід пам'яті подається адреса комірки, в яку потрібно записати дане, і сигналом запису це дане записується у вказану адресою комірку пам'яті. У режимі зчитування на

адресний вхід пам'яті подається адреса комірки, з якої потрібно зчитати дане, і сигналом зчитування це дане зчитується з вказаної адресою комірки пам'яті [1].

Пам'ять з довільним доступом є доволі простою та легкою в використанні. Крім того, її організація задовольняє основні принципи побудови комп'ютерів з архітектурою Джона фон Неймана (в зарубіжній термінології Instruction Set Architecture - ISA), яка є сьогодні базовою під час побудови комп'ютерів [13]. До того ж організація пам'яті з довільним доступом є і основним стримуючим фактором в розвитку комп'ютерів. Це пояснюється низкою її вузьких місць та недоліків, до яких насамперед необхідно зарахувати такі:

- Адресна організація пам'яті з довільним доступом не дозволяє розпаралелити її роботу, зокрема в ній не забезпечується можливість одночасного записування даних з кількох входів в одну комірку, що не дозволяє створити паралельну багатоканальну (багатопортову) пам'ять.
- Необхідність двократного звернення до комірки з даною адресою під час записування числа і при зчитуванні числа сповільнює взаємодію пристроїв комп'ютера з пам'яттю з довільним доступом. І це при тому, що між швидкодією пам'яті та процесора існує дуже великий розрив, який постійно збільшується.
- Потреба звернення до інформації (команд та даних) за адресою вимагає знання місця розташування цієї інформації в пам'яті комп'ютера, що вимагає створення складної системи адресації та відповідних методів взаємодії з нею.
- Необхідність забезпечення доступу до кожної комірки з входу та виходу пам'яті або з об'єднаного входу/виходу пам'яті вимагає великих затрат на елементи доступу та сповільнює швидкодію пам'яті. Причому з ростом ємності пам'яті затрати на елементи доступу зростають, а її швидкодія знижується.

**Пам'ять з асоціативним доступом.** Пам'ять з асоціативним доступом (ПАД) (або асоціативна пам'ять) передбачає зберігання разом з даними і їх ознаки, якою може бути і саме дане. Числа записуються в довільні вільні (звільнені) комірки пам'яті. Дані вибираються з такої пам'яті на основі збігу їх ознак з заданою. У паралельній асоціативній пам'яті одночасно порівнюються з аргументом всі розряди всіх полів ознак пам'яті (так званий одночасний пошук по слову). У послідовній асоціативній пам'яті одночасно порівнюються з бітом аргументу по одному біту кожного поля ознаки (так званий порозрядний пошук). Можливість паралельної роботи є основною перевагою асоціативної пам'яті. З неї можна одночасно зчитати всі дані з однаковими ознаками. Можливість швидкого одночасного перегляду даних у такій пам'яті забезпечує їй широке використання в комп'ютерах, зокрема, за принципом пам'яті з асоціативним доступом часто будується кеш пам'ять [1]. До того ж ця пам'ять також має деякі недоліки, до основних з яких потрібно зарахувати такі:

- При створенні на основі пам'яті з асоціативним доступом багатоканальної (багатопортової) пам'яті необхідно забезпечувати одночасний доступ з кожного каналу до кожної комірки пам'яті, з одночасним порівнянням всіх ознак у комірках з шуканими ознаками, а також забезпечити пошук даних, ознаки яких збіглися з заданою, що є проблематичною задачею, особливо при великих ємностях пам'яті.
- Необхідність двократного звернення до однієї комірки пам'яті під час записування і зчитування числа сповільнює взаємодію пристроїв комп'ютера з пам'яттю з асоціативним доступом.
- Ця пам'ять має доволі складну організацію, яка вимагає необхідності забезпечення доступу до кожної комірки з входу та виходу пам'яті або з об'єднаного входу/виходу пам'яті, причому під час зчитування необхідно порівнювати задану ознаку з ознаками даних у регістрах пам'яті, а також забезпечити пошук даних, ознаки яких збіглися з заданою. Це вимагає великих затрат на елементи доступу та сповільнює роботу пам'яті.

**Пам'ять з послідовним доступом.** Пам'ять з послідовним доступом (ППОД) (англійський термін serial memory) будується так, що дані зчитуються або записуються до цієї пам'яті в послідовному порядку одне за одним, утворюючи деяку чергу [1–5]. Зчитування здійснюється з

черги слово за словом в порядку записування або в зворотному порядку. Прямий порядок зчитування забезпечується пам'яттю типу FIFO з дисципліною обслуговування “перший прийшов – перший вийшов” (First In, First Out). Обернений порядок зчитування забезпечується пам'яттю типу LIFO з дисципліною обслуговування “останній прийшов – перший вийшов” (Last In, First Out), або, що є тим самим, пам'яттю типу FILO з дисципліною обслуговування “перший прийшов – останній вийшов” (First In, Last Out). Пам'ять з послідовним доступом, яка побудована на регістрах, часто називається стеком. Ця пам'ять є швидкою та простою в реалізації.

Недоліком пам'яті з послідовним доступом є значний час доступу до конкретної одиниці інформації. В гіршому випадку для такого доступу може виникнути потреба в перегляді всіх комірок пам'яті.

**Пам'ять з програмованим доступом.** Розглянемо організацію роботи нового типу пам'яті - пам'яті з програмованим доступом (ППРД). Як і пам'ять з послідовним доступом ППРД орієнтована на роботу з масивами даних. У цій пам'яті забезпечується доступ до даних у програмно встановленому порядку, тобто код, який надходить у пам'ять разом з даним, або під час його зчитування, вказує місце даного у вихідному масиві.

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

Робота пам'яті з програмованим доступом організована так [8–12]. Разом із числом в пам'ять надходить код, який вміщує інформацію про подальше використання числа, а саме номер  $N$  числа в масиві, за яким число має з'явитись на виході пам'яті. При багатократному використанні того самого числа можливе використання відповідної кількості кодів  $N$ , або повторний запис числа в пам'ять з новим кодом  $N$ . Всередині пам'яті з програмованим доступом дані запам'ятовуються та проводяться дії з їх переміщення до виходу пам'яті з тим, щоб вони там з'явилися з заданим номером  $N$ . Враховуючи те, що в кожному такті на кожному виході пам'яті з'являється лише одне число, тобто відсутня конфліктна ситуація, немає проблем в реалізації описаних функцій пам'яті. Зрозуміло, що ця пам'ять не може бути використана, коли невідомий номер даного в вихідному масиві.

Порівняно з пам'яттю з довільним, асоціативним та послідовним доступом, пам'ять з програмованим доступом має істотні переваги, основні з яких такі:

- В цій пам'яті наявне тільки однократне звернення за даним, оскільки код номера даного присутній лише під час його записування або зчитування, тоді як в пам'яті з довільним та з асоціативним доступом звернення є двократним. Тим самим в два рази зменшується кількість розрядів в адресній частині команд і, відповідно, спрощується керування та зменшуються затрати обладнання.

- У варіанті пам'яті з програмованим доступом з розділеними входом і виходом забезпечується одночасне записування та зчитування даних, оскільки дані записуються в пам'ять без прив'язки до конкретної її комірки, а зчитуються в заданому порядку, який не пов'язаний з вхідними даними. Тим самим в два рази зростає швидкодія порівняно з пам'яттю з довільним та з асоціативним доступом.

- Немає прив'язки даних до комірок пам'яті, тому відпадає потреба в складних та повільних дешифраторах адрес. Їх функції можуть бути дезінтегровані, що дозволяє істотно зменшити їх затримку та прискорити роботу пам'яті;

- Можуть ефективно вирішуватись питання розпаралелення роботи пам'яті з програмованим доступом, оскільки в ній немає прив'язки даних до комірок пам'яті.

У таблиці наведена інформація стосовно задоволення розглянутими типами пам'яті вимог критеріїв ефективності.

Задоволення типами пам'яті вимог критеріїв ефективності		Типи пам'яті			
		ПДД	ПАД	ППОД	ППРД
1	багатопортовість	-	+	-	+
2	одночасний безконфліктний доступ до даних з багатьох портів	-	+	-	+
3	одночасний запис та зчитування даних	-	-	+	+
4	виконання операцій реорганізації масивів та впорядкування даних в масивах	+	+	-	+
5	відсутність звернення за даними як при їх записі, так і при зчитуванні	-	-	+	+
6	відсутність необхідності звернення до комірки пам'яті при записі і при зчитуванні числа	-	-	+	+
7	відсутність потреби зберігання інформації про місце знаходження даного в пам'яті	-	-	+	+

З аналізу таблиці можна зробити висновок, що найкраще вимогам критеріїв ефективності відповідає пам'ять з програмованим доступом. Це свідчить про доцільність виконання подальших досліджень методів та принципів її побудови та реалізації.

### Висновки

1. Сформовані вимоги до пам'яті та критерії її ефективності під час використання комп'ютера для виконання інтенсивних обчислень над масивами даних.
2. Розглянуто класифікацію та проаналізовано пам'ять сучасних комп'ютерів.
3. Описані принципи організації та характеристики різних типів пам'яті, визначено та обгрунтовано переваги та вузькі місця і недоліки кожного типу пам'яті.
4. Проаналізовано задоволення існуючими типами пам'яті вимог критеріїв ефективності. Показано, що найкраще вимогам критеріїв ефективності відповідає пам'ять з програмованим доступом.

1. Мельник А.О. *Архітектура комп'ютера*. – Луцьк: Волинська обласна друкарня, 2008. – 470 с. 2. D. Patterson, J. Hennessy. *Computer Architecture. A Quantitative Approach*. Morgan Kaufmann Publishers, Inc. 1996. 3. Stallings, W. *Computer Organization and Architecture, 5th ed.*, New York, NY: Macmillan Publishing Company, 2000. 4. Tannenbaum, Andrew. *Structured Computer Organization, 4th ed.*, Upper Saddle River, NJ: Prentice Hall, 1999. 5. Узрюмов Е.П. *Цифровая схемотехника*. – СПб.: БХВ – Санкт-Петербург. 2000. – 528с. 6. Мельник А.О. *Програмовані процесори обробки сигналів*. – Львів. „Львівська політехніка, 2000. - 57 с. 7. Шугин А.А., Дерюгин А.А. *Процессоры и память*. 8. А.с. 1529287 (СССР). *Запоминающее устройство* / А.А. Мельник. – Опубл. в Б.И., 1989, № 46. 9. А.с. 1479954 (СССР). *Буферное запоминающее устройство* / А.А. Мельник. – Опубл. в Б.И., 1989, № 18. 10. Мельник А.О. *Запам'ятовуючі пристрої сортувального типу для систем цифрової обробки сигналів. Матер. Першої Всеукр. конф. „Обробка сигналів і зображень та розпізнавання образів”*. – К., 17–21 листопада 1992 р. – С. 187–188. 11. Мельник А.О. *Принципи побудови буферної сортувальної пам'яті* // Вісн. Держ. ун-ту “Львівська політехніка” “Комп'ютерна інженерія та інформаційні технології”, 1996. – №307. – С.65–71. 12. Мельник А.О. *Спеціалізовані комп'ютерні системи реального часу*. – Львів: Держ. ун-т “Львівська політехніка”, 1996. – 54 с. 13. Burks, A. W., H. H. Goldstine, and J. von Neumann [1946]. “Preliminary discussion of the logical design of an electronic computing instrument,” Report to the U.S. Army.