

Розділ 9

Багаторівнева пам'ять комп'ютера

Пам'ять є одним із основних вузлів комп'ютера, що призначений для зберігання інформації, тобто програм і даних. Функції пам'яті забезпечуються запам'ятовуючими пристроями, які здійснюють приймання, зберігання і видачу інформації в процесі роботи комп'ютера. Процес приймання інформації в запам'ятовуючий пристрій називають записом, процес видачі інформації – зчитуванням, а спільно їх визначають як процеси звернення до пам'яті.

В даному розділі будуть розглянуті питання побудови пам'яті з довільним, впорядкованим та асоціативним доступом. Кожний тип пам'яті має свої переваги та недоліки, які визначають місце його використання в комп'ютері. Буде розглянута структура багаторівневої пам'яті та її типи, які входять до складу внутрішньої та зовнішньої пам'яті комп'ютера а також основні характеристики пам'яті: ємність, організація, швидкодія, час доступу, період звернення, вартість. Зокрема буде проведено аналіз можливих варіантів організації регістрових файлів процесорів та обмежень, які спричиняє використання багаторівневого регістрового файла, а також розглянуто ряд нових структур регістрового файла: інтегрованого, розподіленого кластерного, з керованою комутацією, з віконною організацією, ієрархічного. Буде розглянута динамічна та статична організація даних в регістрових файлах та описана робота регістрового файла на базі чергі з програмовою затримкою.

Будуть також розглянуті принципи роботи пам'яті з асоціативним доступом та описано чотири основні елементи її організації: з повним паралельним асоціативним доступом, з неповним паралельним асоціативним доступом, з послідовним асоціативним доступом, з частково асоціативним доступом.

Буде наведено опис варіантів побудови основної пам'яті та можливості по скороченню часу доступу до інформації. Зокрема буде розглянута блокова структура пам'яті, принципи розшарування пам'яті та нарощування її розрядності. Буде описано роботу різних типів постійного запам'ятовуючого пристрою.

При розгляді зовнішньої пам'яті буде наведений порядок розміщення інформації на магнітному диску, типи сучасних дискових систем, в тому числі масиви магнітних дисків з надлишковістю, оптична пам'ять та пам'ять на магнітних стрічках.

9.1. Типи та характеристики пам'яті комп'ютера

9.1.1. Багаторівнева структура пам'яті комп'ютера

Пам'ять комп'ютера є багаторівневою. На різних рівнях в комп'ютері використовуються різні типи пам'яті (рис. 9.1). Практично кожен функціональний вузол комп'ютера має у своєму складі пам'ять. Так, в процесорі знаходиться регістрова надоперативна пам'ять, а також постійна пам'ять для зберігання мікропрограм, яка є складовою частиною пристрою керування, та постійна пам'ять констант, яка використовується в АЛП, наприклад, у складі табличних або таблично-алгоритмічних операційних пристрій. Між процесором і основною пам'яттю може бути включено кілька рівнів кеш пам'яті. Крім основної пам'яті, в комп'ютері є ще пам'ять процесорів введення-виведення та зовнішня пам'ять великого об'єму, реалізована на магнітних дисках, стрічках та барабанах, а також оптична та флеш. Як видно з рисунку, за місцем розташування пам'яті поділяють на внутрішню і зовнішню. Найбільш швидкісну пам'ять, тобто кеш-пам'ять першого рівня, зазвичай розміщують на одному кристалі з центральним процесором, в якому є своя регістрова пам'ять. До внутрішньої пам'яті належать також основна пам'ять, пам'ять процесора введення-виведення та кеш пам'ять другого і подальших рівнів (кеш пам'ять другого рівня може також розміщуватися на кристалі процесора). Повільну пам'ять великої ємності (магнітні й оптичні диски, магнітні стрічки та барабани) називають зовнішньою пам'яттю, оскільки до ядра комп'ютера ці пристрій підключаються аналогічно до пристрій введення-виведення.



Рис. 9.1. Багаторівнева структура пам'яті комп'ютера

9.1.2. Типи пам'яті

В комп'ютері використовуються різні типи пам'яті, які, залежно від способу доступу до інформації, можуть бути класифіковані наступним чином:

■ Пам'ять з довільним доступом. До цієї пам'яті в кожному такті може бути записане число або зчитане з неї число за довільною адресою. За принципом пам'яті з довільним доступом побудовано регістровий файл регістрової пам'яті процесора та основна пам'ять.

■ Пам'ять із впорядкованим доступом. Із такої пам'яті дані вибираються в порядку, який визначається внутрішньою структурою пам'яті. До такої пам'яті належить зокрема пам'ять з послідовним доступом, з якої дані зчитуються послідовно одне за одним. За принципом пам'яті з послідовним доступом побудовано, зокрема, буферну пам'ять, яка використовується у пристроях введення-виведення, зовнішню пам'ять на магнітних стрічках.

■ Пам'ять з асоціативним доступом. В такій пам'яті дані шукаються за їх змістом або за деякою їх ознакою. За принципом пам'яті з асоціативним доступом побудовано, зокрема, кеш пам'ять.

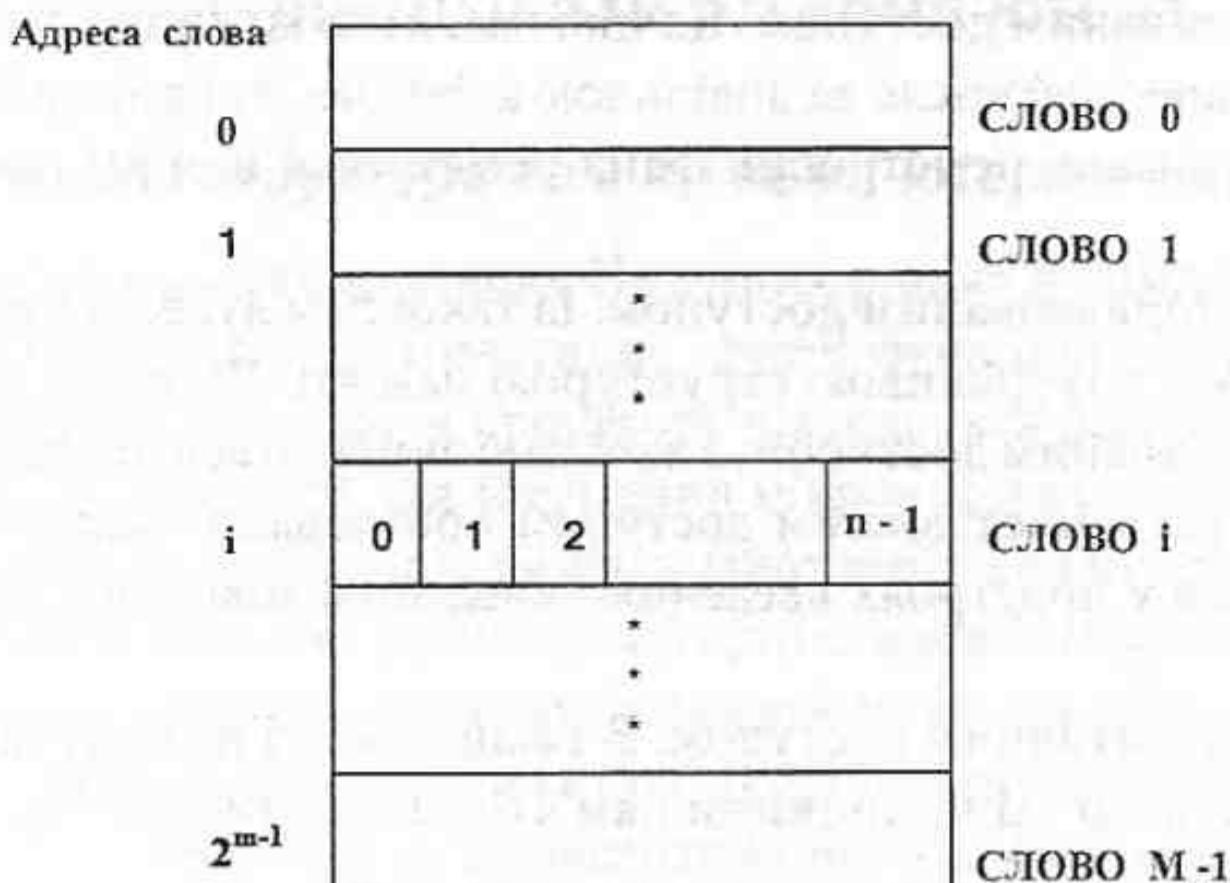
Існує також пам'ять з прямим доступом. Кожен запис має унікальну адресу, що відображає її фізичне розміщення на носії інформації. Звернення здійснюється як адресний доступ до початку запису з подальшим послідовним доступом до певної одиниці інформації усередині запису. В результаті час доступу до певної позиції є величиною змінною. Такий режим характерний для магнітних дисків.

Цей список може бути розширеній іншими типами пам'яті, які ще не знайшли широкого застосування в комп'ютері, наприклад, пам'яттю з програмованим доступом, коли записи до пам'яті дані зчитуються в наперед заданому порядку. Кожний тип пам'яті має свої переваги та недоліки, які визначають місце використання відповідної пам'яті в комп'ютері. Розглянемо організацію роботи та проведемо аналіз названих типів пам'яті детальніше.

Найчастіше в комп'ютері використовується пам'ять з довільним доступом або адресна пам'ять. Ця пам'ять ділиться 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).

Пам'ять із довільним доступом складається з комірок, кожна з яких зберігає одиницю інформації, яка називається словом. Слови складаються із бітів із 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 ж саму адресу має і слово, яке зберігається в даній 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 $M-1$ (рис. 9.2). M адрес є адресним полем (простором) даного комп'ютера. Використовуючи двійкове кодування, необхідно m бітів для представлення всіх адрес, де $m = [\log_2 M]$. Значення в дужках означає більше ціле. Зазвичай пам'ять комп'ютера будується 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^m . Це значення вказує також ємність пам'яті в бітах.

Рис. 9.2. Адресна пам'ять організацією $n \cdot 2^m$

Пам'ять з довільним доступом виконує дві основні операції: вибірку Fetch (або зчитування Read) і запам'ятування Store (або запис Write). Робота цієї пам'яті організована наступним чином. В режимі запису 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. В 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 (рис. 9.3).



Рис. 9.3. Інтерфейс адресної пам'яті

Інтерфейс адресної пам'яті (рис. 9.3 а) включає m -розрядну шину адреси, n -розрядну шину вхідних даних, n -розрядну шину вихідних даних та однорозрядний вхід задання режиму роботи: запису або зчитування. Зазвичай використовуються також вхід дозволу використання пам'яті та вхід для подачі тактових імпульсів, необхідних для синхронізації роботи пам'яті. Слід зауважити, що у варіанті інтерфейсу адресної пам'яті, представлена на рис. 9.3 а, шини вводу та виводу даних розділені. Часто пам'ять будується з об'єднаними шинами вводу та виводу даних (тобто шина даних тут є двонаправленою) (рис. 9.3 б), що спрощує використання такої пам'яті в комп'ютерах із одношинною та багатошинною структурами.

Пам'ять з довільним доступом може бути реалізована у вигляді оперативної або постійної пам'яті, та у вигляді стека програмно-доступних регістрів (регистрового файла). В останньому випадку (рис. 9.4) адреса вказує номер регістра. Тут ДША – дешифратор

адреси. Розрядність адреси дорівнює $m = [\log_2 k]$, де k – кількість регістрів. Як відомо, регістр – це найшвидший запам'ятовуючий елемент комп'ютера. Він використовується для короткотермінового зберігання одного слова інформації. Регістр будується на основі тригерів, кожен з яких зберігає найменшу одиницю інформації біт, тобто до нього може бути записано 0 або 1. Для зберігання n -розрядного слова регістр повинен мати n тригерів.

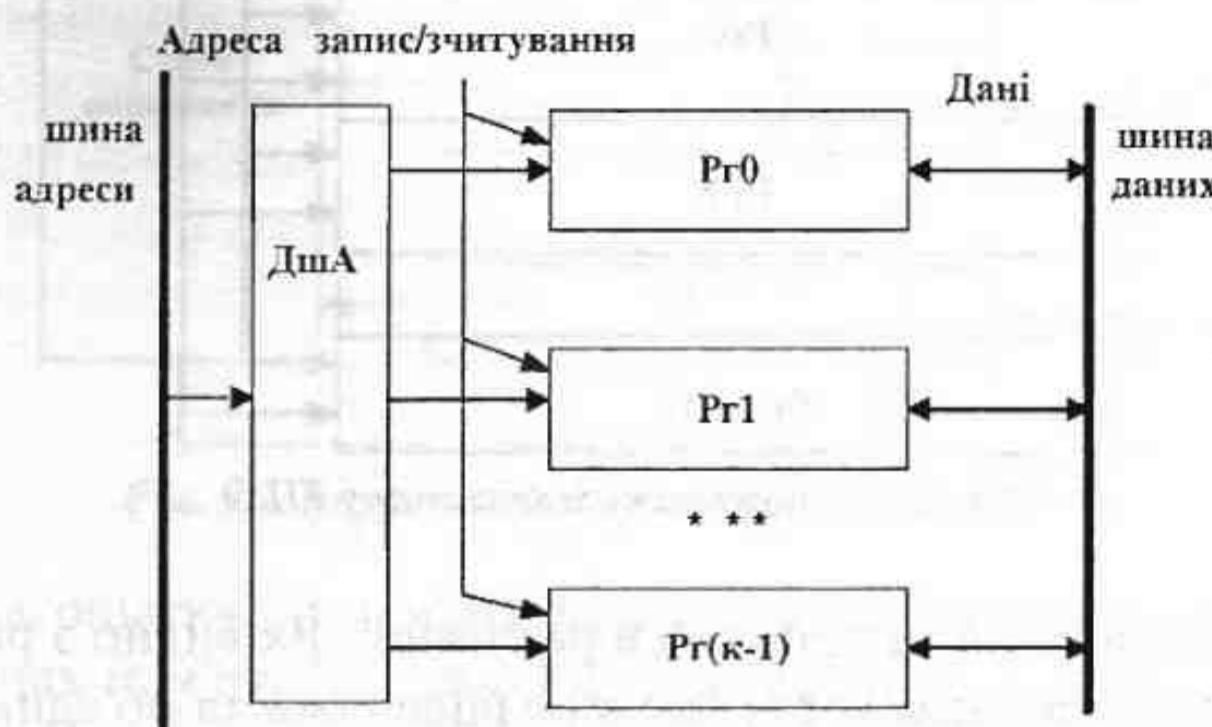


Рис. 9.4. Стек програмно-доступних регістрів

В режимі запису інформація записується лише в вибраний дешифратором адреси ДША регістр. А в режимі зчитування на шину даних поступає лише інформація з вибраного дешифратором адреси ДША регістра. Виходи інших регістрів в цей час знаходяться в стані високого імпедансу.

З пам'яті з впорядкованим доступом дані вибираються в порядку, який визначається її внутрішньою структурою. Пам'ять з впорядкованим доступом будується таким чином, що дані вибираються, або записуються, через вхідний регістр, а правило їх переміщення між комірками пам'яті "зашите" в зв'язках між ними.

До такої пам'яті належить зокрема пам'ять з послідовним доступом, широко використовувана в комп'ютерах, з якої дані зчитуються послідовно одне за одним в порядку їх запису або в порядку зчитування.

Пам'ять з послідовним доступом будується таким чином, що дані зчитуються або записуються до цієї пам'яті в послідовному порядку одне за одним, утворюючи деяку чергу. Зчитування здійснюється з черги слово за словом в порядку запису 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 же 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. Приклад такої пам'яті показано на рис. 9.5, де дані з шини через регістр даних РгД поступають до послідовно з'єднаних регістрів пам'яті Pr0, Pr1, ... Pr(k-1), по яких рухаються вверх або вниз залежно від значення сигналу режиму роботи.

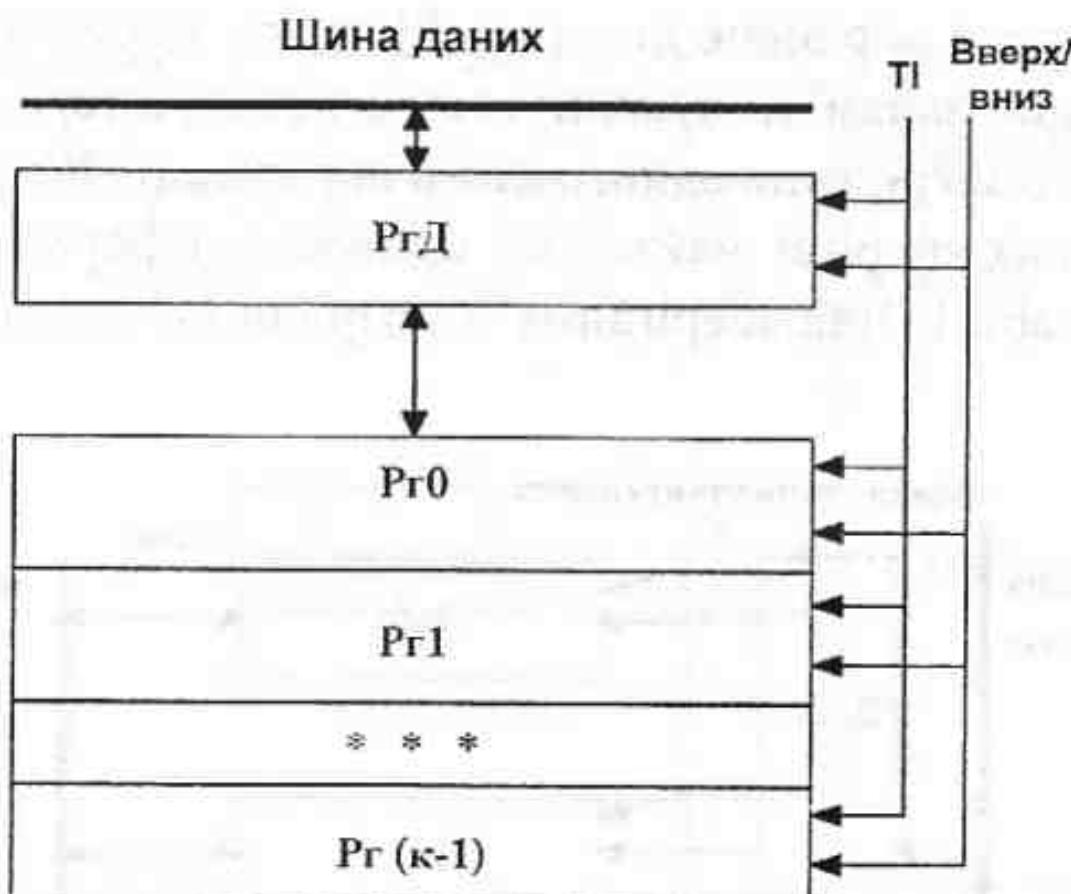


Рис. 9.5. Стекова пам'ять типу FILO

Ця пам'ять є найшвидшою та простою в реалізації. Як видно з рис. 9.6, де показано інтерфейс пам'яті з послідовним доступом з розділеними та об'єднаними входом і виходом, в цій пам'яті відсутній адресний вхід.

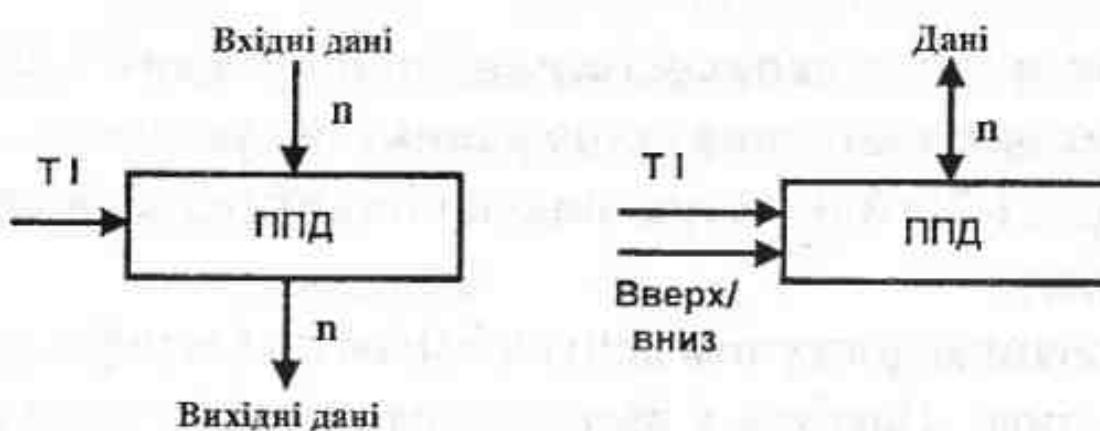


Рис. 9.6. Інтерфейс пам'яті з послідовним доступом з розділеними та об'єднаними входом і виходом

Пам'ять з розділеними входом і виходом може одночасно приймати вхідні дані та видавать попередньо прийняті дані, які рухаються з входу до виходу за допомогою тактових імпульсів T1. Коли ж вхід і вихід об'єднані, то дані вводяться в пам'ять тактовими імпульсами, які поступають по входу режиму роботи "зверх", а виводяться з пам'яті тактовими імпульсами, які поступають по входу режиму роботи "вниз".

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 всього об'єму пам'яті.

Пам'ять з асоціативним доступом (або асоціативна пам'ять) зберігає разом з даними і їх ознаки, в ролі якої може бути і саме дане. Ядро пам'яті з асоціативним доступом показане на рис. 9.7. Роль комірок цієї пам'яті виконують регистри. Числа записуються в довільні вільні регистри пам'яті. Дані вибираються з такої пам'яті на основі збігу їх ознак з заданою. Для цього ознаки даних з усіх регистрів пам'яті поступають на схему порівняння, де порівнюються з заданою ознакою із регистра ознаки, і на вихід пам'яті поступають дані, ознаки яких збігаються з заданою.

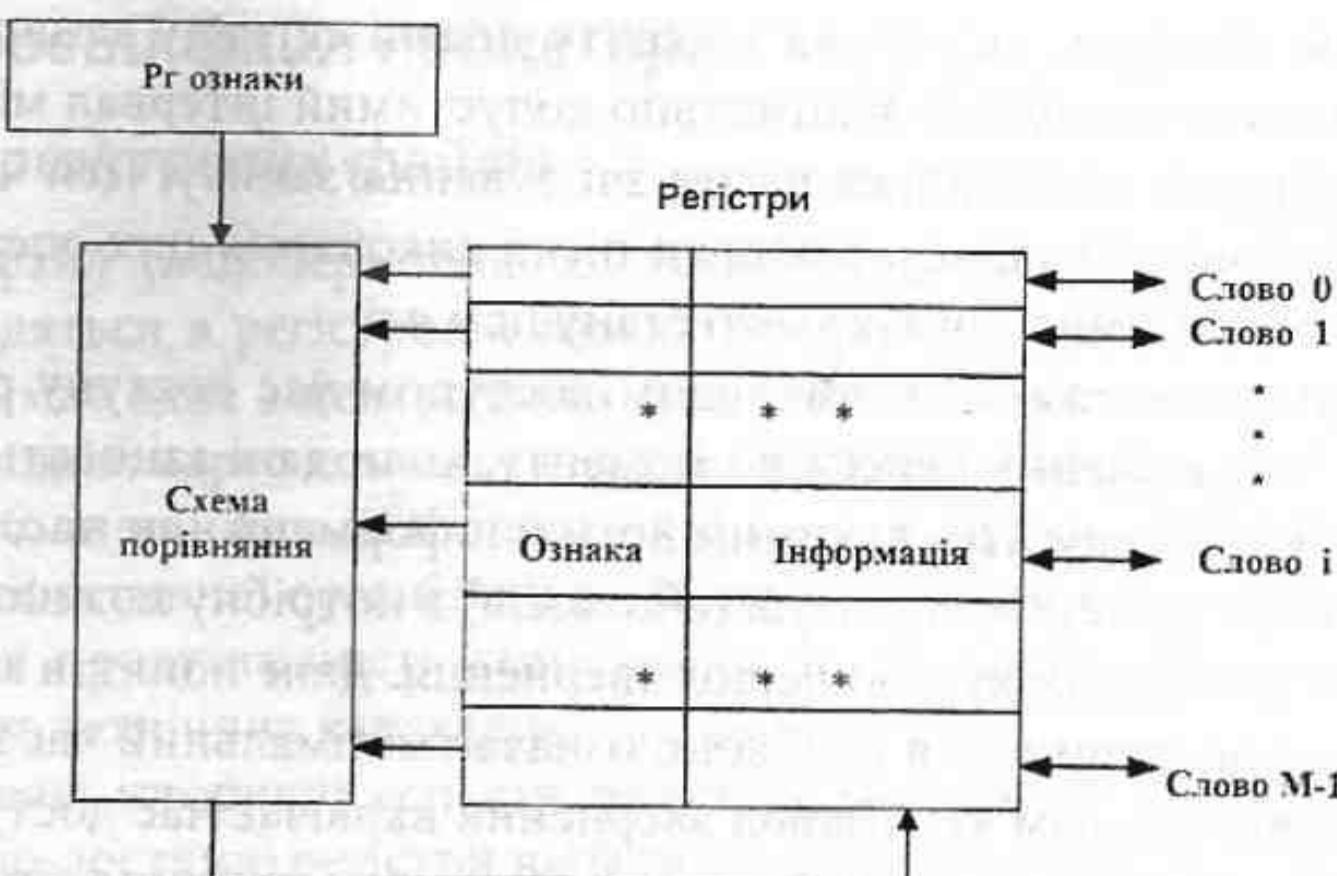


Рис. 9.7. Ядро пам'яті з асоціативним доступом

В паралельній асоціативній пам'яті одночасно порівнюються з аргументом всі ряди всіх полів ознак пам'яті (так званий одночасний пошук по слову). В 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 одинаковими ознаками. Можливість швидкого одночасного перегляду даних в такій пам'яті забезпечує її широке використання в комп'ютерах, зокрема за принципом пам'яті з асоціативним доступом часто будується кеш пам'ять.

Пам'ять буває енергозалежною та енергонезалежною. У енергозалежній пам'яті інформація може бути спотворена або втрачена при відключенні джерела живлення. У енергонезалежній пам'яті записана інформація зберігається і при відключенні на пруги живлення. Магнітна і оптична пам'ять є енергонезалежними. Напівпровідникова пам'ять може бути як енергозалежною, так і енергонезалежною.

Залежно від технології виготовлення розрізняють наступні основні типи пам'яті: напівпровідникова пам'ять, пам'ять з магнітним носієм інформації, використовувана в магнітних дисках і стрічках, та пам'ять з оптичним носієм – оптичні диски.

9.1.3. Основні характеристики пам'яті

До основних характеристик пам'яті належать: ємність, організація, швидкодія та вартість.

- Ємність виражають числом бітів або байтів, які можуть зберігатися в пам'яті. Добавлення до одиниці виміру множника К означає множення на $2^{10} = 1024$, множника М – множення на $2^{20} = 1048576$, а множника Г – множення на $2^{30} = 1073741824$.

Важливою характеристикою пам'яті є її організація, тобто кількість комірок в пам'яті $M = 2^m$ та розрядність кожної комірки n . Хоча організація пам'яті вказує на її ємність, це є самостійна характеристика, оскільки при тій же ємності організація пам'яті може бути різною.

Для кількісної оцінки швидкодії пам'яті використовують наступні характеристики:

- Час зчитування та час запису, або час зчитування/запису. Час зчитування – це інтервал часу між моментами подачі сигналу зчитування та появи слова на виході пам'яті.

ті. Час запису – це інтервал часу після моменту подачі сигналу запису, достатній для запам'ятування слова в пам'яті. Мінімально допустимий інтервал між послідовними зчитуванням та записом називається часом зчитування/запису. Цей час може перевищувати час зчитування або запису, оскільки після виконання цих операцій може бути потрібний час на відновлення початкового стану пам'яті.

- Час доступу. Для пам'яті з довільним доступом час доступу рівний інтервалу часу від моменту надходження адреси до моменту, коли дані заносяться в пам'ять або стають доступними. Для пам'яті з рухомим носієм інформації – це час, що витрачається на установку головки запису/зчитування (або носія) в потрібну позицію.

- Тривалість циклу пам'яті або період звернення. Дане поняття застосовується до пам'яті з довільним доступом, для якої воно означає мінімальний час між двома послідовними зверненнями до пам'яті. Період звернення включає час доступу плюс деякий додатковий час. Додатковий час може бути потрібен для затухання сигналів на лініях, а в деяких типах пам'яті, де зчитування інформації приводить до її руйнування, – для відновлення зчитаної інформації.

Вартість пам'яті прийнято оцінювати відношенням її загальної вартості до ємності в бітах або байтах, тобто вартістю зберігання одного біта або байта інформації. На рис. 9.8 показана зміна з роками вартості зберігання одного МВ інформації та часу доступу до одиниці інформації в нс для трьох типів пам'яті: статичної напівпровідникової пам'яті з довільним доступом (SRAM), динамічної напівпровідникової пам'яті з довільним доступом (DRAM) та пам'яті на магнітних дисках (Disk).

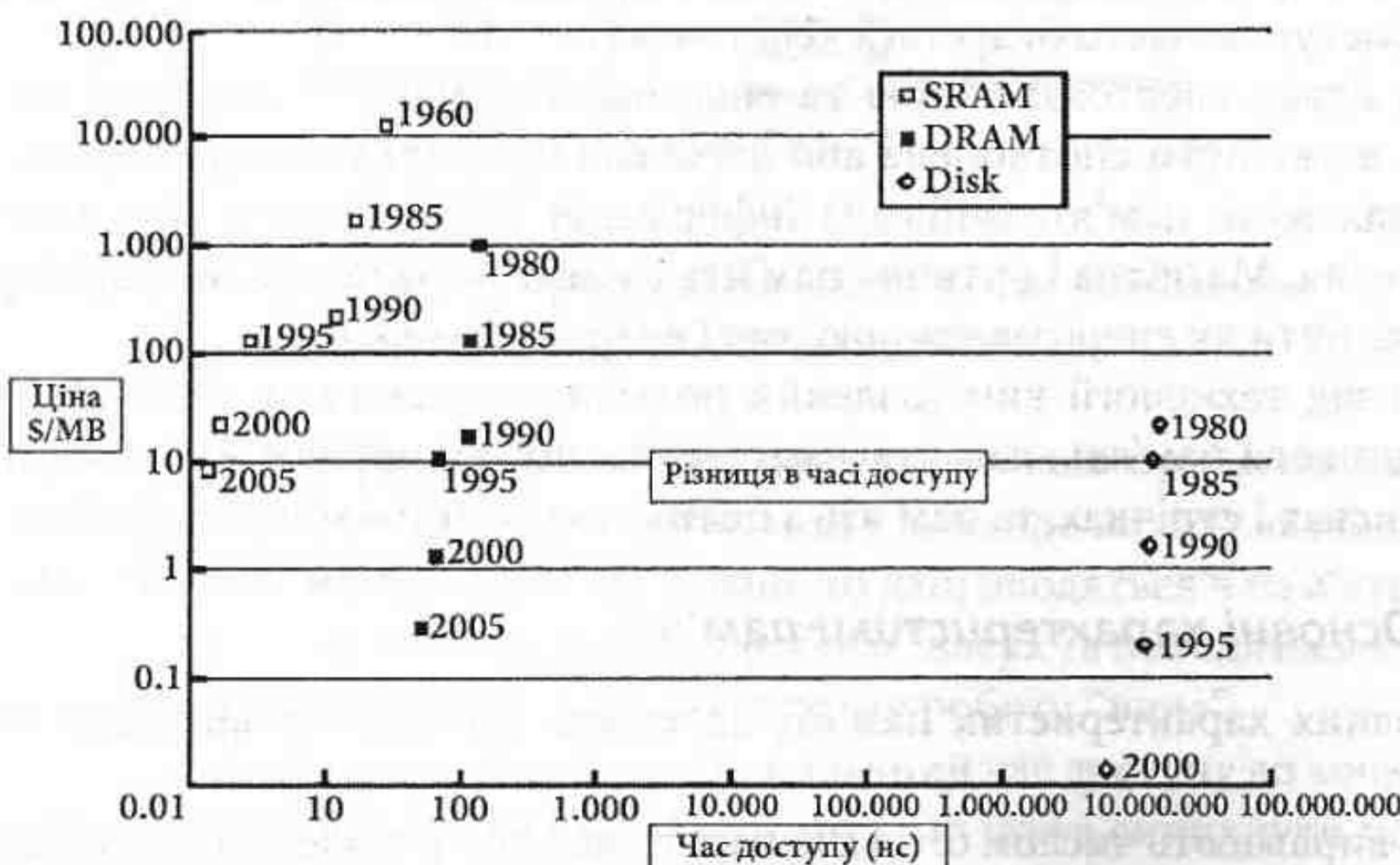


Рис. 9.8. Зміна з роками вартості зберігання одного МВ інформації та часу доступу до одиниці інформації

Як бачимо, ціна зберігання одного МВ інформації в статичній пам'яті залишається досить високою (більше, ніж в 10 разів вищою, ніж в динамічній пам'яті). Але, і час доступу до даних в цій пам'яті є значно меншим (майже в 100 разів). При цьому, як видно з рисунку, зберігається суттєвий розрив між ціною зберігання одного МВ інформації та часом доступу до даних в статичній та динамічній пам'яті, і дисковій пам'яті.

9.2 Регістровий файл процесора

9.2.1. Типи регістрових файлів

Регістровий файл (надоперативна пам'ять) – це набір програмно-доступних реєстрів, які знаходяться в регістровій пам'яті процесора. В сучасних процесорах регістровий файл займає одне з центральних місць. Він використовується для локального збереження операндів, адрес команд та даних, індексів, а також дозволяє організовувати ефективний обмін даними між операційними пристроями процесора та основною пам'яттю. Вибір ефективної організації регістрового файла є одним із підходів, що дозволяє підвищити продуктивність комп'ютера. Як правило, організація регістрового файла належить до технічних характеристик комп'ютера.

Розглянемо деякі історичні аспекти розвитку організації регістрового файла процесора. Програмно-доступні реєстри почали використовуватися з початку 1960-х років. В 1964 році фірма IBM розробила серію універсальних комп'ютерів IBM/360 для наукових та комерційних розрахунків, в процесорах яких був використаний регістровий файл, що включав 16 32-розрядних цілочисельних реєстрів та 16 64-розрядних реєстрів з рухомою комою. Також в 1964 році був виготовлений перший суперкомп'ютер для наукових розрахунків CDC 6600, процесор якого мав регістровий файл, що включав 24 реєстри. В 1977 році був спроектований перший векторний суперкомп'ютер Cray-1, регістровий файл якого мав ієрархічну структуру. Його було поділено на файл основних реєстрів і файл другорядних (фонових) реєстрів. Кількість основних реєстрів була меншою, що дозволяло забезпечити швидкий доступ до них, тоді як кількість другорядних реєстрів була більшою, проте доступ до них був повільнішим. Операнди тривалого зберігання розміщувалися в другорядних реєстрах і переміщувалися в основні реєстри при потребі їх використання. Cray-1 загалом містив 656 реєстрів даних і адрес (включаючи векторні реєстри, але не враховуючи реєстрів керування). На особливу увагу заслуговує група з восьми векторних реєстрів. Кожний такий реєстр міг вміщувати 64-х елементний вектор з рухомою комою. Однією 16-ти розрядною командою можна було додати, відняти чи перемножити два вектори.

З наведених прикладів видно, що для організації високопродуктивних обчислень важливо не тільки мати велику кількість реєстрів, але і забезпечити швидкий доступ до даних, які в них зберігаються, за рахунок ефективної організації регістрового файла.

На рис. 9.9 приведена класифікація типів регістрових файлів процесора. В них використовується статична організація збереження даних, тобто при записі і читанні даних вказується безпосередня адреса реєстра, яка є незмінною під час виконання програми.

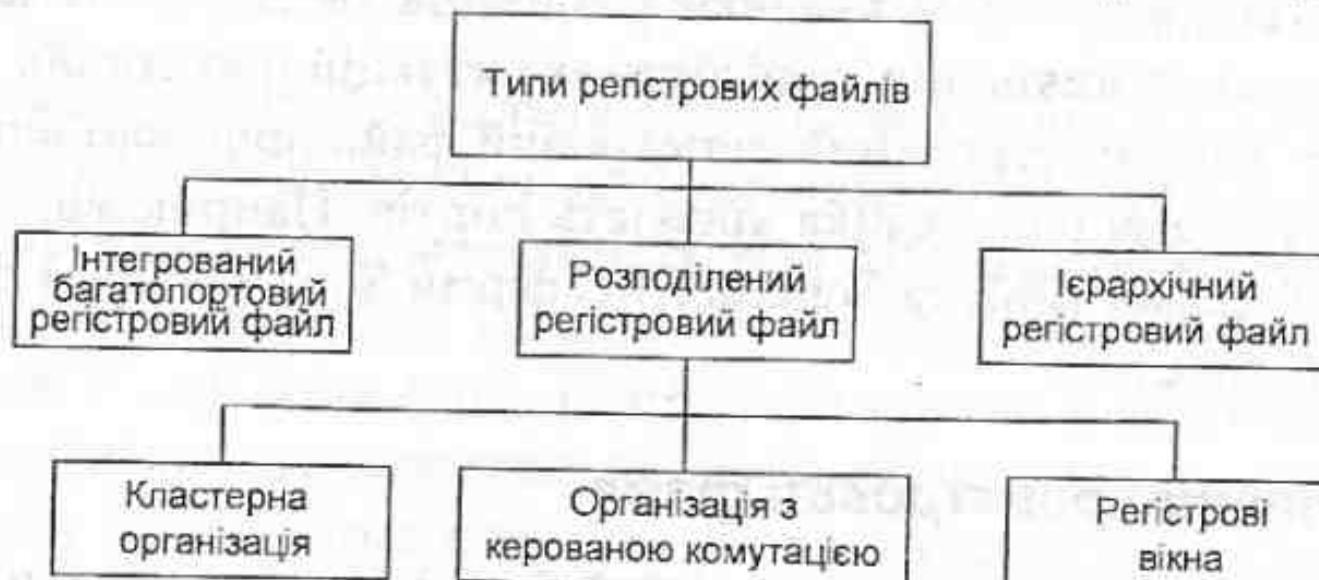


Рис. 9.9. Класифікація типів регістрових файлів

Розглянемо наведені на рис. 9.9 типи регістрових файлів детальніше.

9.2.2. Інтегрований багатопортовий регістровий файл

В інтегрованому багатопортовому регістровому файлі забезпечується доступ до будь-якого регістру з кожного його порта. На рис. 9.10 показана структура інтегрованого регістрового файла з двома портами зчитування (rd_1 , rd_2), та одним портом запису (wd), який містить 32 32-розрядних регістри (R_0-R_{31}). В регістр R_0 запис заборонений, оскільки в ньому зберігається константа нуль. Вихід кожного регістра під'єднаний до відповідних інформаційних входів мультиплексорів МП1 та МП2. Адреси (rs_1 , rs_2), що подаються на входи керування мультиплексорів, визначають номери регістрів, з яких читаються дані. Для запису використовується демультиплексор ДМП, через який на відповідний регістр подається сигнал запису даних (we), що поступає з вхідної шини, та адреса регістра ws . Процес запису синхронізується тактовими імпульсами, що подаються на вхід clk . Кожний з регістрів має вхід дозволу запису даних En , інформаційний вхід D , та інформаційний вихід Q .

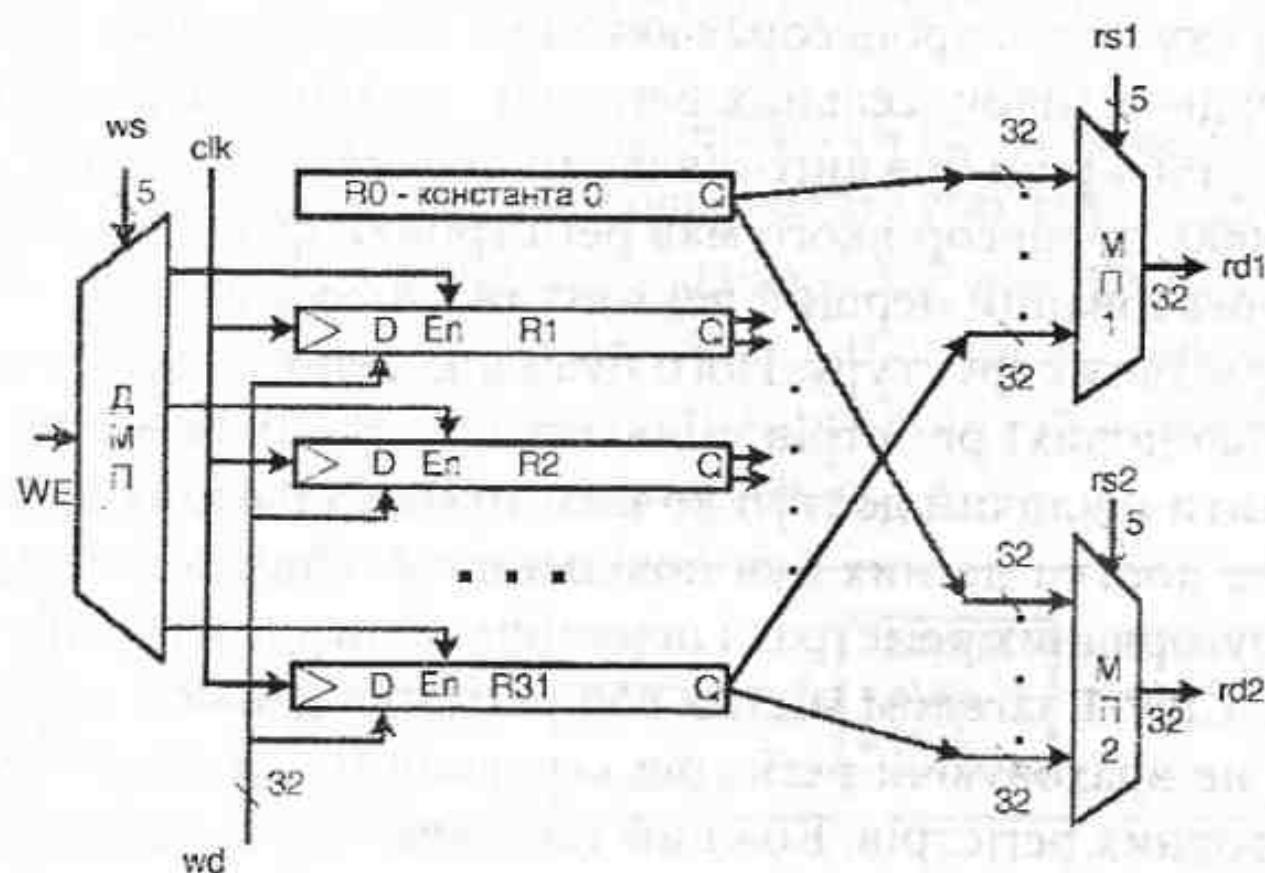


Рис. 9.10. Структура інтегрованого регістрового файла

Із збільшенням кількості регістрів регістрового файла зростає час доступу до даних, так як ускладнюються мультиплексори та демультиплексори, а також відповідно збільшується розмір кристалу процесора.

Збільшення кількості портів зчитування та запису призводить до збільшення кількості відповідно мультиплексорів та демультиплексорів, а також суттєво збільшує кількість шин передачі даних, що ускладнює реалізацію регістрового файла. Крім того, з'являється можливість виникнення конфліктних ситуацій при спробі запису даних до тих самих регістрів. Тому інтегрований регістровий файл доцільно використовувати в комп'ютерах, де не вимагається велика кількість портів. Наприклад, регістрові файли комп'ютерів RS/6000 фірми IBM, та SuperSPARC фірми Sun, містять 4 порти для зчитування і 2 порти для запису.

9.2.3. Розподілений регістровий файл

Використання розподілених регістрових файлів дозволяє зменшити площину кристала, яку займає регістрова пам'ять процесора. Розподілення регістрового файла здійснюється шляхом поділу функціональних елементів процесора на групи, кожна з яких має свій

локальний регістровий файл. Відповідно, такі регістрові файли містять меншу кількість портів та регістрів, що зменшує затрати обладнання на їх реалізацію. Як ми вже бачили, існує три типи розподілених регістрових файлів: кластерні, з керованою комутацією та з віконною організацією. Розглянемо принципи їх побудови.

9.2.3.1. Кластерний розподілений регістровий файл

Локальні регістрові файли можуть бути повністю незалежними, тобто дані для певної групи функціональних елементів процесора доступні тільки з конкретного локального регістрового файла. Такий розподілений регістровий файл, який складається з незалежних локальних регістрових файлів, відповідно до наведеної вище класифікації, називають кластерним.

Кластерні регістрові файли здебільшого використовуються у векторних процесорах. В таких процесорах одна операція над всіма п компонентами векторних операндів задається однією командою. Кожен з таких векторних операндів міститься в локальному регістровому файлі біля відповідного функціонального елемента процесора, і відпадає необхідність у повноцінних зв'язках між локальними регістровими файлами.

На рис. 9.11 показана кластерна організація регістрового файла процесора TMS320C64x фірми Texas Instruments. В цьому процесорі функціональні елементи поділені на підмножину А (L_1, S_1, M_1, D_1) та підмножину В (L_2, S_2, M_2, D_2). Функціональні елементи процесора L (L_1, L_2) виконують арифметичні операції та операції порівняння. Функціональні елементи процесора S (S_1, S_2) виконують арифметично-логічні операції та команди керування. Функціональні елементи процесора M (M_1, M_2) виконують множення 16-ти розрядних операндів, а функціональні елементи процесора D (D_1, D_2) виконують арифметичні операції, та виконують роль генераторів адрес.

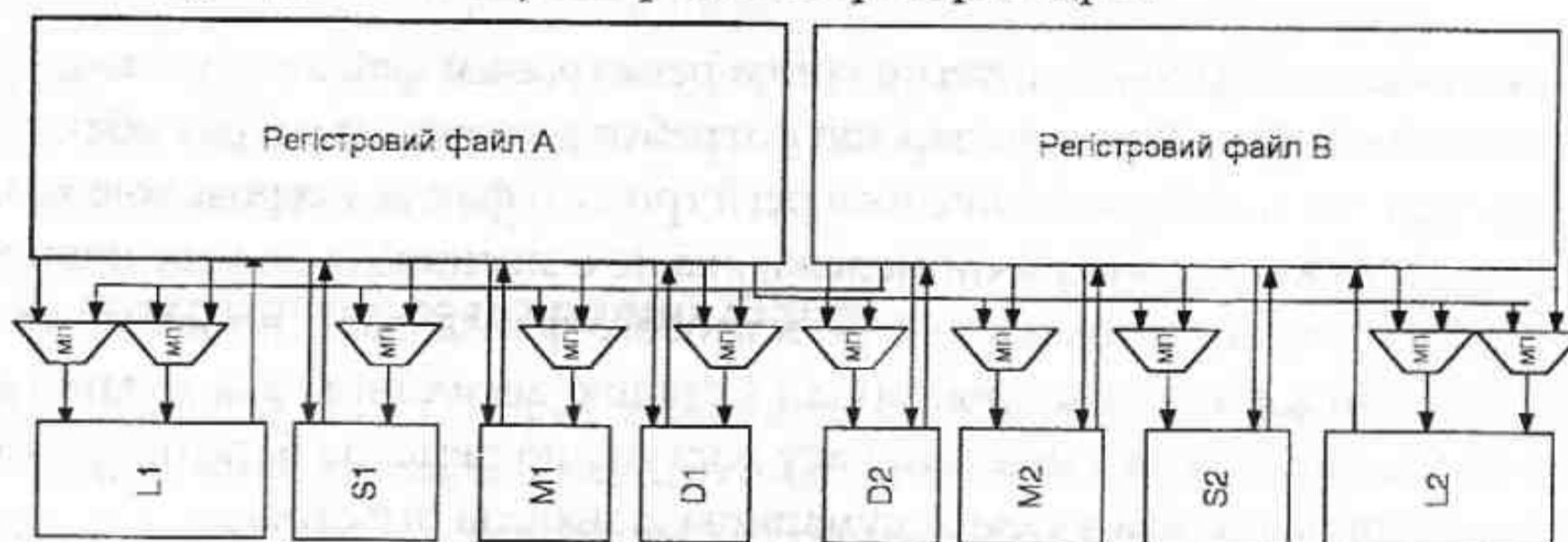


Рис. 9.11. Організація регістрового файла процесора TMS320C64x фірми Texas Instruments

Кожна підмножина має свій локальний регістровий файл. Для пересилання даних з регістрового файла однієї підмножини до функціональних елементів іншої підмножини використовуються додаткові мультиплексори МП. Фактично, можливість повноцінної пересилки обмежена, оскільки для цього в кожному з локальних регістрових файлів виділяється тільки один порт. Саме з цієї причини організацію регістрового файла TMS320C64x можна назвати кластерною. Зрозуміло, що описана організація регістрового файла для приведеного типу процесорів дозволяє зменшити затрати обладнання та прискорити час доступу до регістрів у порівнянні з інтегрованим регістровим файлом того ж об'єму.

9.2.3.2. Розподілений регістровий файл з керованою комутацією

Якщо будь-якій групі функціональних елементів процесора доступні дані з будь-якого локального регістрового файла, то це буде інший варіант організації розподіленого регістрового файла. Його називають розподіленим регістровим файлом з керованою комутацією.

Розподілений регістровий файл з керованою комутацією, структура якого приведена на рис. 9.12, складається з локальних регістрових файлів (ЛРФ), що зв'язані з функціональними елементами (ФЕ) процесора через комутуючу мережу, керування якою виконує контролер комутуючої мережі (ККМ).



Рис. 9.12. Розподілений регістровий файл з керованою комутацією

Тут для кожного входу функціонального елемента ФЕ процесора виділяється окремий локальний регістровий файл ЛРФ з одним портом для зчитування та одним портом для запису даних. Тому кількість локальних регістрових файлів є в два рази більшою кількості функціональних елементів процесора. Об'єм (кількість регистрів) кожного локального регістрового файла є сталим, і не залежить від кількості функціональних елементів.

Порівняно з багатопортовим інтегрованим регістровим файлом тут додатково введена комутуюча мережа та її контролер, що потребує додаткових затрат обладнання.

Час доступу до регистрів розподіленого регістрового файла з керованою комутацією складається з затримки в комутуючій мережі, яка не є значною із-за невеликої кількості входів, та із часу доступу до локального регістрового файла, який не залежить від кількості функціональних елементів процесора і є сталою величиною для заданої кількості регистрів. В сумі ця величина є меншою часу доступу до регистра багатопортового інтегрованого регістрового файла з тією ж сумарною кількістю регистрів.

9.2.3.3. Розподілений регістровий файл з віконною організацією

В розглянутих структурах розподілених регістрових файлів доступ до регистрів для обчислювальних модулів розмежовувався в просторі. Існує інша організація розподілених регістрових файлів, яка передбачає розподілений доступ до регистрів в часі. Така організація отримала назву регістрових вікон. В основному вона використовується для зменшення довжини команди та споживаної потужності процесора, а також дає можливість нарощувати кількість регистрів без зміни формату команди. За затратами обладнання та часом доступу розподілений регістровий файл із віконною організацією близький до розподіленого регістрового файла з керованою комутацією. На рис. 9.13 показано структуру розподіленого регістрового файла з віконною організацією. Тут регістровий

файл поділений на N вікон, кожному з яких відповідає один з registrovих файлів РФ₀ – РФ_{N-1}. Для зміни активного вікна використовується сигнал вибору файла з блоку керування.

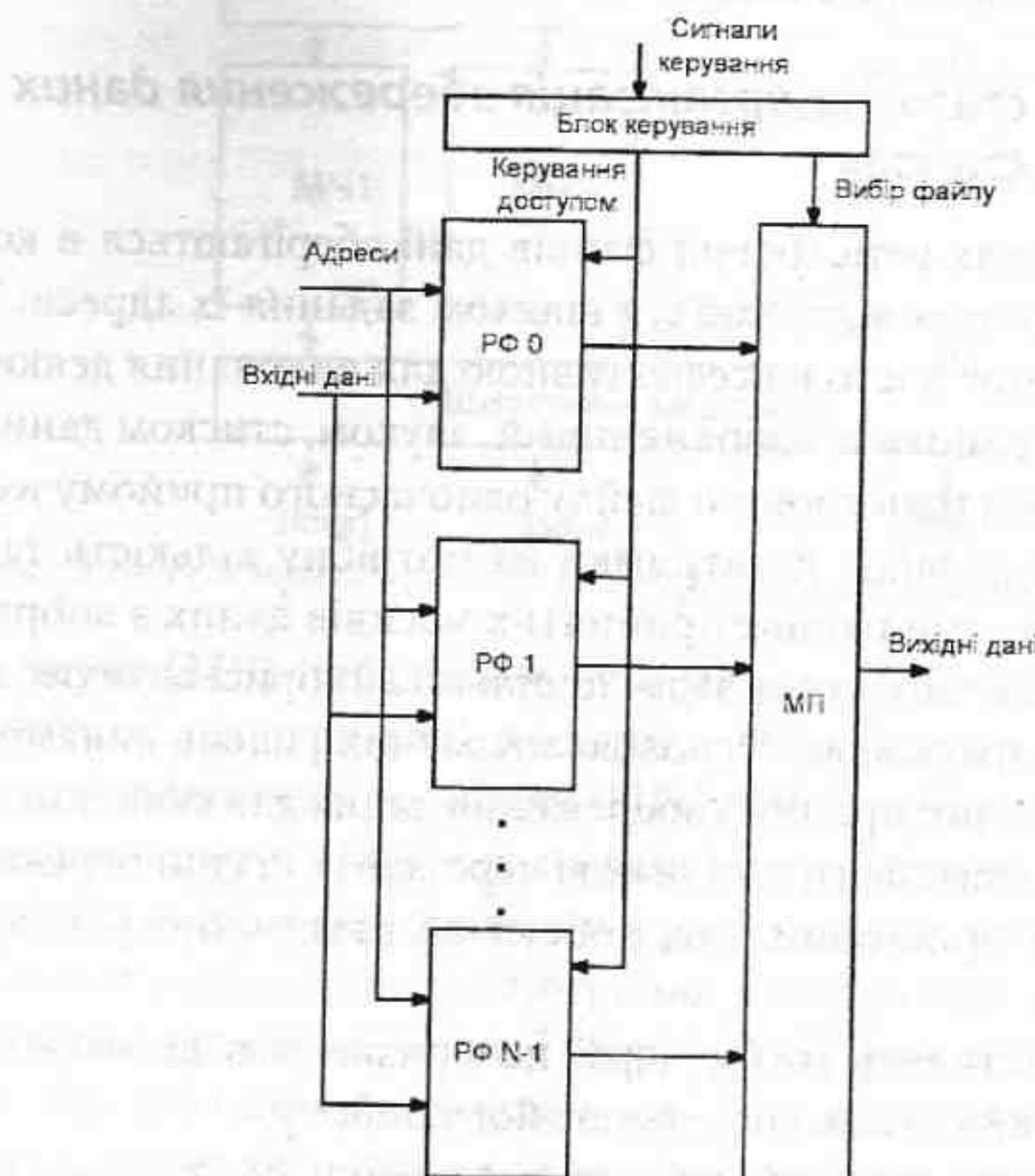


Рис. 9.13. Розподілений registrovий файл з віконною організацією

Потрібно зазначити, що організація registrovого файла у вигляді registrovих вікон є неефективною для систем із високим рівнем паралелізму, оскільки вона дозволяє збільшити кількість registrov, але не дозволяє організовувати паралельний доступ до будь-якого registrov в один момент часу.

9.2.4. Ієрархічний registrovий файл

В універсальних комп’ютерах важливим є той факт, що звернення процесора до пам’яті завжди локалізовано в невеликому діапазоні змін її адрес. Саме він дозволяє застосовувати ієрархічну організацію системи пам’яті, аби розв’язати невідповідність швидкодії процесора та пам’яті, яка передбачає введення кількох рівнів кеш пам’яті та ефективну організацію обміну інформацією між цими рівнями. Однак існує множина алгоритмів з великими наборами входних даних, що характеризуються дуже низьким рівнем повторного використання тих же комірок пам’яті. При виконанні таких алгоритмів кеш пам’ять є зайвою ланкою в системі пам’яті комп’ютера. В цьому випадку для того, щоб більш ефективно реалізувати доступ до основної пам’яті, будуються ієрархічні registrovі файли шляхом поділу інтегрованого registrovого файла на registrovий файл великого об’єму з декількома портами для виконання операцій з пам’яттю і registrovий файл меншого об’єму з великою кількістю портів для обслуговування функціональних елементів процесора. Такий поділ дозволяє зменшити затрати обладнання на реалізацію ієрархічного registrovого файла у порівнянні з інтегрованим.

Регістровий файл процесора SPARC фірми Sun можна з деякими застереженнями віднести до ієрархічних регістрових файлів, оскільки крім віконної організації він містить асоціативну регістрову кеш пам'ять.

9.2.5. Динамічна та статична організація збереження даних в регістрових файлах

В розглянутих структурах регістрових файлів дані зберігаються в конкретних реєстрах. Доступ до цих реєстрів відбувається шляхом задання їх адреси. Така статична організація збереження даних часто є неефективною для виконання деяких класів алгоритмів. Наприклад, задачі роботи із зображеннями, звуком, стиском даних в реальному масштабі часу вимагають від регістрового файла одночасного прийому вхідних масивів даних з декількох вхідних каналів, їх затримки на потрібну кількість тактів та видачі декількома вихідними каналами раніше прийнятих масивів даних з впорядкуванням за заданим законом. Для вирішення таких задач доцільно використовувати динамічне збереження даних в регістрових файлах. Основою апаратних рішень динамічних регістрових файлів є черги. Динамічний принцип збереження даних для складних задач обробки інформації в реальному масштабі часу на основі черг має наступні переваги:

- доступ до черги є простіший, ніж доступ до статичного регістрового файла, оскільки відсутні адреси;
- збільшення кількості реєстрів у черзі не призводить до вагомого зростання складності регістрового файла та не сповільнює його роботу;
- простота оптимізації структури регістрового файла під заданий алгоритм;
- необхідна менша кількість адресних ліній, порівняно зі статичними регістровими файлами;
- вирішується проблема розподілу реєстрів, оскільки дані записуються не в конкретний реєстр, а в чергу.
- можливість реалізації регістрового файла на основі модулів напівпровідникової пам'яті з довільним доступом.

Слід зауважити наступну статистику: 40 % результатів виконання арифметичних операцій використовуються на наступному такті виконання програми, а ще 45 % – через такт. Такі дані ще раз переконують в ефективності реалізації регістрових файлів на основі черг.

Регістровий файл на основі черг може бути побудований за принципом пам'яті з послідовним доступом типу FIFO. Недоліком такої організації є те, що для алгоритмів, в яких умови сумісності черг не виконуються, різко збільшуються затрати обладнання на його реалізацію.

Кращою є реалізація регістрового файла на основі черг, побудованого на пам'яті з програмованою затримкою. Структура такого регістрового файла, який можна віднести до розподілених регістрових файлів із керованою комутацією та динамічним збереженням даних, наведена на рис. 9.14. Такий тип структури має найбільші можливості врахування особливостей алгоритмів, що дозволяє виконувати алгоритми з максимальною продуктивністю.



Рис. 9.14. Регістровий файл на базі черги з програмованою затримкою

Тут модуль черги (МЧ) побудовано на основі двопортової адресної пам'яті. Один порт дозволяє читати дані з пам'яті, а інший – записувати дані в пам'ять. Зчитування даних виконується аналогічно до пам'яті FIFO, тобто через кожний тактовий імпульс читається комірка за адресою, більшою на одиницю від попередньої. Збільшення адреси забезпечується лічильником. Алгоритм просування даних в модулі представляє собою генерування сигналу зчитування для потрібної комірки пам'яті даного модуля. Таким чином кожна комірка є пронумерована і зберігає свій номер. Лічильник пам'яті генерує послідовні значення, що визначають номер комірки, яка повинна видавати дані назовні. Отже, як бачимо, система зчитування дуже проста – дані зчитуються тоді, коли приходить їх черга. А впорядкування здійснюється на етапі запису: до комірки якого номера запищуться дані, на такті з тим номером вони і видадуться назовні. Адреса запису формується за допомогою лічильника та суматора шляхом додавання значення затримки (адреса А) та значення лічильника. Двх_i – i-й вхідний порт даних, Двих_i – i-й вихідний порт даних ($i=1,2,\dots,k$).

9.3. Пам'ять з асоціативним доступом

9.3.1. Організація та типи пам'яті з асоціативним доступом

Пам'ять з асоціативним доступом (або асоціативна пам'ять) широко використовується в сучасних комп'ютерах. У першу чергу за принципом пам'яті з асоціативним доступом побудовано запам'ятовуючі пристрої кеш пам'яті більшості комп'ютерів. Крім того, за принципами пам'яті з асоціативним доступом побудовано буферну пам'ять в конвеєрних процесорах, зокрема це буфер попередньої вибірки команд, буфер впорядкування і т. д. Асоціативна пам'ять ефективна також для побудови пристрій для виконання операцій числового пошуку (min, max, сортування і т. д.).

Пам'ять з асоціативним доступом передбачає зберігання разом із даними і їх ознаки, в ролі якої може бути і саме дане. Дані вибираються з такої пам'яті на основі збігу їх ознак із заданою. Тому цю пам'ять іще називають пам'яттю, що адресується за вмістом або за даними. Ядро пам'яті з асоціативним доступом приведене на рис. 9.7. На рис. 9.15 приведена детальніша структура одного з варіантів побудови пам'яті з асоціативним доступом.

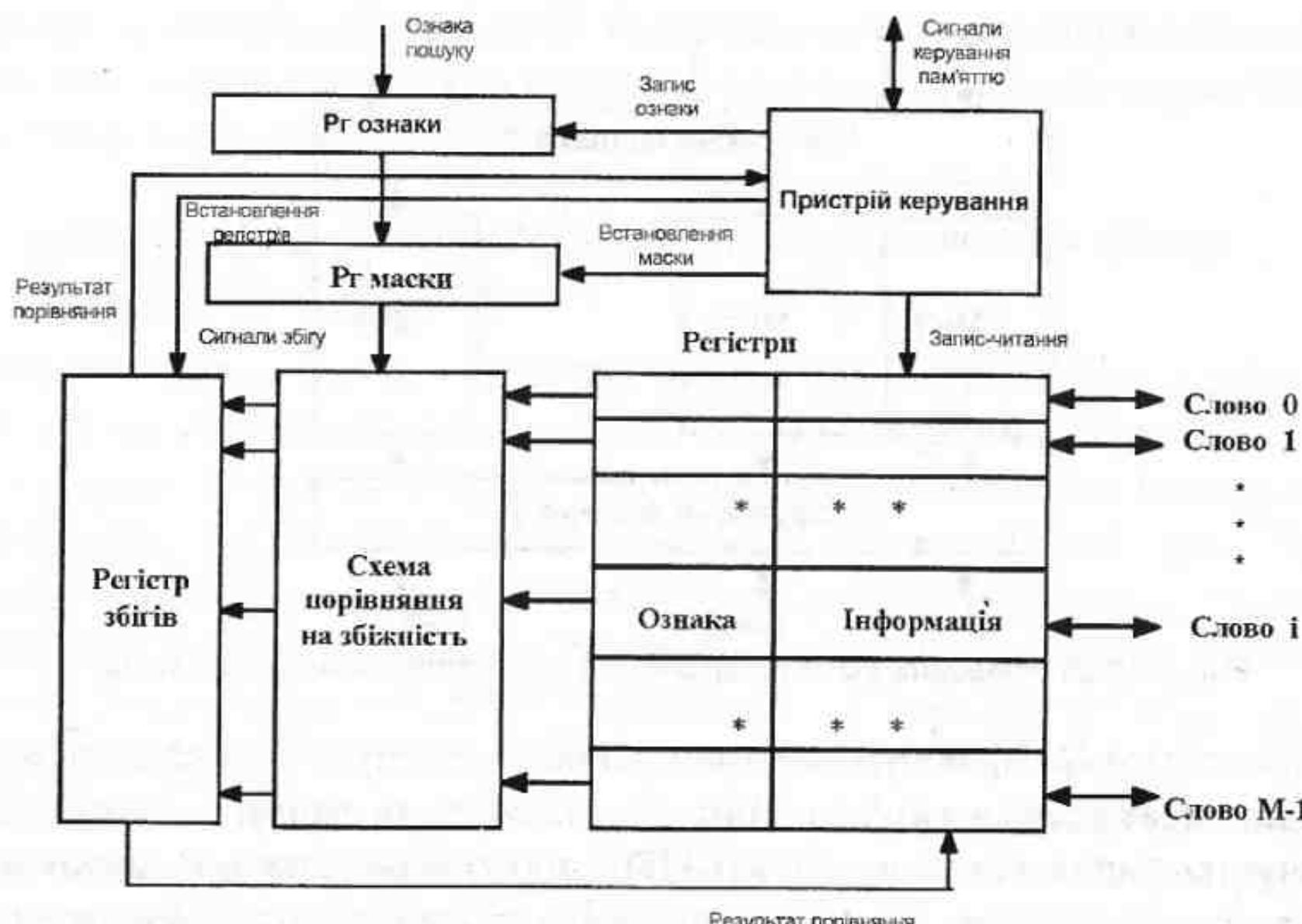


Рис. 9.15. Структура пам'яті з асоціативним доступом

Ця пам'ять включає:

- масив регістрів для зберігання N m-роздрядних слів, в кожному з яких частина розрядів зайнята ознаками, а іншу частину розрядів займає службова інформація;
- регістр асоціативної ознаки, куди поміщається код шуканої інформації (ознака пошуку). Розрядність цього регістра зазвичай є меншою чи рівною довжині слова n;
- схему порівняння на збіжність, яка використовується для проведення паралельного або послідовного порівняння бітів ознак пошуку всіх слів, що зберігаються, з відповідними бітами асоціативної ознаки пошуку, і вироблення сигналів збігу;
- регістр збігів, в якому кожному регістру масиву пам'яті відповідає один розряд, в який заноситься одиниця, якщо всі розряди ознак слова відповідного регістра співпали з однайменними розрядами асоціативної ознаки пошуку;
- регістр маски, який дозволяє заборонити порівняння певних бітів;
- пристрій керування, який на основі зовнішніх сигналів керування здійснює запис асоціативної ознаки пошуку до відповідного регістра, встановлює маску в регістрі маски, та на основі даних порівняння на збіжність з регістра збігів здійснює зчитування даних із регістрів асоціативної пам'яті, а також запис до них входних даних.

При зверненні до пам'яті з асоціативним доступом, спочатку сигналами з блоку керування розряди регістра маски, які не повинні враховуватися при пошуку інформації, встановлюються в нульові значення, і всі розряди регістра збігів встановлюються в одиничний стан. Після цього в регістр асоціативної ознаки заноситься код шуканої інформації (ознака пошуку) і починається її пошук, в процесі якого схема порівняння на збіжність одночасно порівнює перший біт ознак даних всіх регістрів пам'яті, з першим бітом асоціативної ознаки пошуку. Ті схеми, які зафіксували неспівпадіння, формують сигнал, що переводить відповідний біт регістра збігів у нульовий стан. Так само відбувається процес пошуку і для решти незамаскованих бітів ознаки пошуку.

У результаті, одиниці зберігаються лише в тих розрядах реєстра збігів, які відповідають реєстрам, де знаходитьсья шукана інформація. Конфігурація одиниць в реєстрі збігів використовується в якості адрес, за якими проводиться зчитування даних із пам'яті.

Внаслідок того, що результати пошуку можуть мати кілька варіантів, вміст реєстра збігів подається на пристрій керування, де формуються сигнали сповіщення про результати порівняння, а саме про те, що шукана інформація не знайдена, міститься в одному чи кількох реєстрах. Тому, при зчитуванні спочатку проводиться аналіз результатів порівняння. Потім, при наявності інформації про те, що шукана інформація не знайдена, зчитування відміняється, при повідомленні, що шукана інформація міститься в одному реєстрі, зчитується слово, на яке вказує одиниця в реєстрі збігів, а при повідомленні, що шукана інформація міститься в кількох реєстрах, скидається найстарша одиниця в реєстрі збігів і витягується відповідне їй слово. Шляхом повторення цієї операції послідовно зчитуються всі слова.

Запис у пам'ять з асоціативним доступом проводиться без вказівки конкретної адреси, в перший вільний реєстр. Для пошуку вільного реєстра виконується операція зчитування, в якій не замасковані тільки службові розряди, що показують, як давно проводилося звернення до кожного реєстра, і вільним вважається або порожній реєстр, або той, який найдовше не використовувався.

Головна перевага пам'яті з асоціативним доступом визначається тим, що час пошуку інформації залежить тільки від числа розрядів в означені пошуку і швидкості опиту розрядів, і не залежить від числа реєстрів пам'яті.

Разом з тим, ця пам'ять також має ряд недоліків, до основних з яких слід віднести наступні:

- Необхідність двократного звернення до однієї комірки пам'яті при записі і при зчитуванні числа в два рази сповільнює взаємодію пристрій комп'ютера з пам'яттю з асоціативним доступом в порівнянні з варіантом пам'яті, в якій можливе лише одне звернення.
- Ця пам'ять має досить складну організацію, яка вимагає забезпечення доступу доожної комірки з входу та виходу пам'яті або з об'єднаного входу-виходу пам'яті, причому при зчитуванні необхідно проводити порівняння заданої ознаки з ознаками даних в реєстрах пам'яті, а також забезпечити пошук даних, ознаки яких співпадали із заданою. Це вимагає великих затрат на елементи доступу та сповільнює роботу пам'яті.
- При створенні на основі пам'яті з асоціативним доступом багатоканальної (багатопортової) пам'яті необхідно забезпечувати одночасний доступ із кожного каналу до доожної комірки пам'яті, з одночасним порівнянням всіх ознак у комірках із шуканими ознаками, а також забезпечити пошук даних, ознаки яких співпадали із заданою, що є проблематичним завданням, особливо при великих об'ємах пам'яті.

Базуючись на концепції асоціативного пошуку можна побудувати цілу низку структур пам'яті з асоціативним доступом. Конкретна структура визначається тим, як поєднані наступні чотири основні елементи її організації:

- вид пошуку інформації;
- спосіб зчитування інформації при множинних збігах;
- спосіб запису інформації;
- варіант порівняння ознак.

У кожному конкретному застосуванні пам'яті з асоціативним доступом завдання пошуку інформації може формулюватися по різному.

При простому пошуку, який був розглянутий вище, потрібен повний збіг всіх розрядів ознак пошуку з однотипними розрядами ознак слів, що зберігаються в реєстрах пам'яті.

При складному пошуку можуть ставитись наступні завдання:

- Пошуку слів із максимальним або мінімальним значенням ознаки. При цьому багаторазова вибірка з пам'яті слова з максимальним або мінімальним значенням ознаки (з виключенням його з подальшого пошуку), по суті, є впорядкованою вибіркою інформації.
- Пошук слів, ознака яких порівняно з асоціативною ознакою пошуку є найближчим більшим або меншим значенням. Ця операція також забезпечує проведення впорядкованої вибірки інформації.
- Знаходження слів, чисельне значення ознаки в яких є більшим або меншим заданої величини. Подібний підхід дозволяє вести пошук слів з ознаками, що лежать всередині або поза заданими межами.

Очевидно, що реалізація складних методів пошуку вимагає внесення відповідних змін у структуру пам'яті з асоціативним доступом.

Важливим питанням є організація зчитування з пам'яті з асоціативним доступом інформації, коли з асоціативною ознакою пошуку співпадають ознаки декількох слів. У цьому випадку застосовується один із двох підходів: використовується ланцюг черговості або проводиться опитування. Використання ланцюга черговості дозволяє проводити зчитування слів у порядку зростання номера реєстра пам'яті незалежно від величини асоціативних ознак. При проведенні опитувань впорядковуються ті розряди, які були замасковані, і не брали участі в пошуку. В іншому варіанті для цих цілей виділяються спеціальні розряди реєстрів.

Істотні відмінності в структурі пам'яті з асоціативним доступом можуть бути пов'язані з вибраним принципом запису інформації. Раніше був описаний варіант із записом у незайнятий реєстр з найменшим номером. На практиці застосовуються й інші способи, наприклад, за співпадінням ознак, за чергою та інші. Найскладнішим є запис із сортуванням інформації на вході пам'яті з асоціативним доступом за величиною ознаки слова. Тут місце положення реєстра, куди буде поміщене нове слово, залежить від співвідношення ознаки записуваного слова і ознак слів, що вже зберігаються в пам'яті з асоціативним доступом.

При побудові пам'яті з асоціативним доступом вибирають один з чотирьох варіантів порівняння ознак, тобто організації перегляду вмісту пам'яті: паралельно за всіма словами, послідовно за всіма словами, паралельно за всіма розрядами, послідовно за всіма розрядами. Ці варіанти можуть комбінуватися. Названі чотири варіанти порівняння ознак визначають принципи побудови схеми порівняння на збіжність. Відповідно до цих принципів, може бути запропоновано кілька типів організації пам'яті з асоціативним доступом. У пам'яті з повним паралельним асоціативним доступом одночасно порівнюються всі розряди ознак всіх слів пам'яті (так званий одночасний пошук по слову). Це найшвидша та найскладніша з точки зору затрат обладнання структура пам'яті з асоціативним доступом. Можливе порівняння не всіх розрядів ознак слів пам'яті, а по частинах, аж до порозрядного порівняння, коли одночасно порівнюються з бітом асоціативної ознаки пошуку по одному біту ознаки кожного слова (так званий порозрядний

пошук). Це буде структура пам'яті з неповним паралельним асоціативним доступом та з послідовною обробкою розрядів ознак слів. Можливе також проведення порівняння одночасно ознак не всіх слів, а по групах. Тоді це буде структура пам'яті з послідовним асоціативним доступом із порозрядним та паралельним порівнянням.

Розглянемо названі чотири основні типи організації пам'яті з асоціативним доступом детальніше.

9.3.2. Пам'ять з повним паралельним асоціативним доступом

Перший тип організації пам'яті з асоціативним доступом – пам'ять з повним паралельним асоціативним доступом. Прикладом конкретного технічного виконання пам'яті такого типу може бути пам'ять комп'ютерної системи PEPE фірми Bell Laboratories. На рис. 9.16 показана загальна структура пам'яті з повним паралельним асоціативним доступом. Пам'ять складається з деякої кількості N комірок, в яких зберігаються дані. Кожна комірка може зберігати слово з деякою кількістю p двійкових розрядів (ширина слова). На рис. 9.17 приведена будова комірки цієї пам'яті. Кожна така комірка може зберігати дані та порівнювати їх з ознакою пошуку паралельно з іншими комірками. Тут і далі розглядається випадок, коли з ознакою пошуку порівнюються всі розряди слова, хоча зрозуміло, що може бути взята і його частина. Дані та сигнали керування надходять від пристрою керування до кожної комірки пам'яті паралельно.

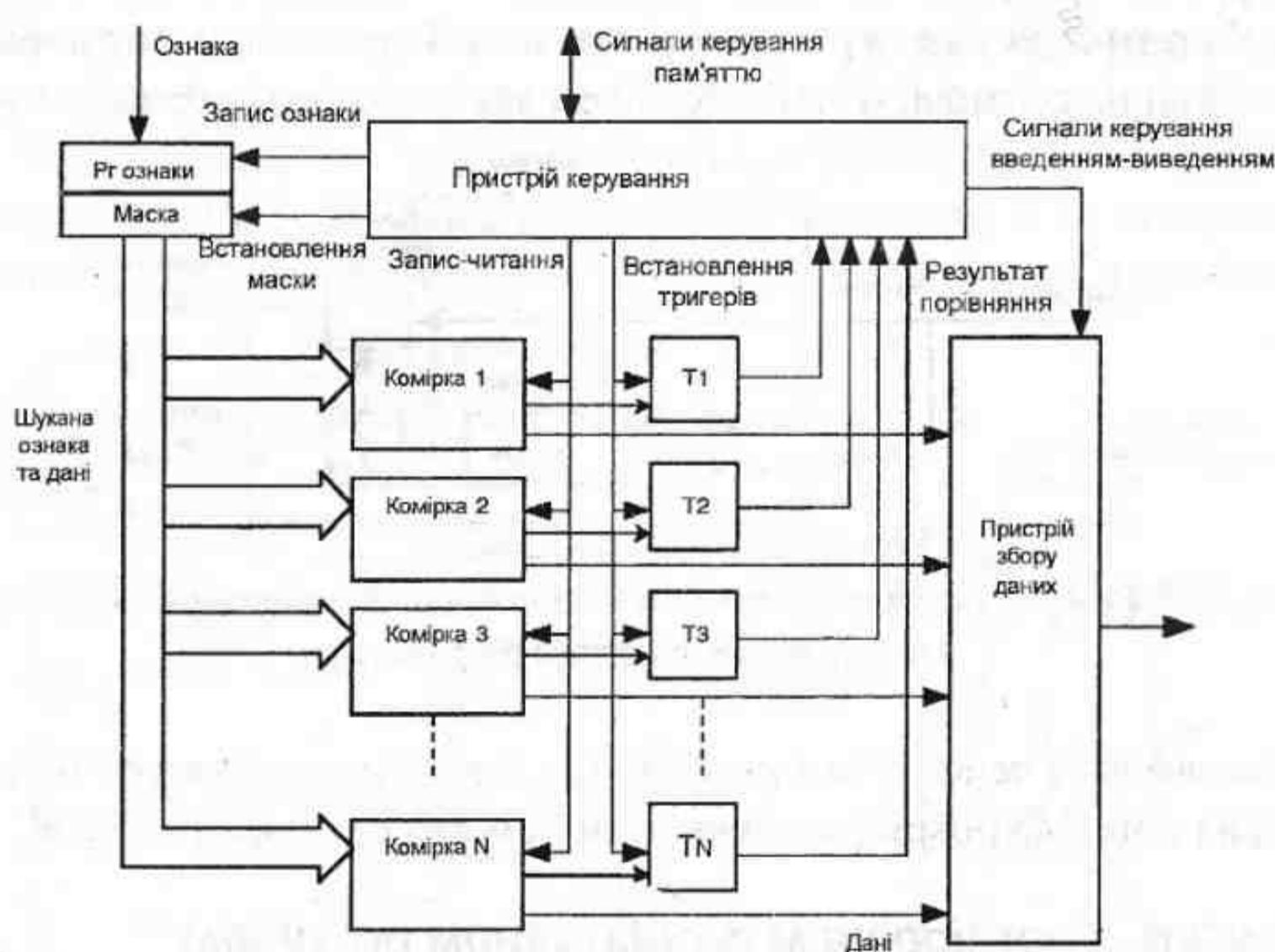


Рис. 9.16. Структура пам'яті з повним паралельним асоціативним доступом

Кожна комірка пов'язана з пристроєм керування теговим розрядом T_i . Перед початком порівняння пристрій керування надсилає команду встановлення всіх тегових розрядів в "1". Коли пристрій керування формує команду порівняння, то кожна комірка, в якій міститься слово, ознака якого не збігається з ознакою пошуку, формує сигнал, який викликає скид відповідного тегового розряду в "0". Після виконання даної команди встановленими в "1" залишається тільки ті розряди, які відповідають коміркам, де відбувся збіг з ознакою пошуку.

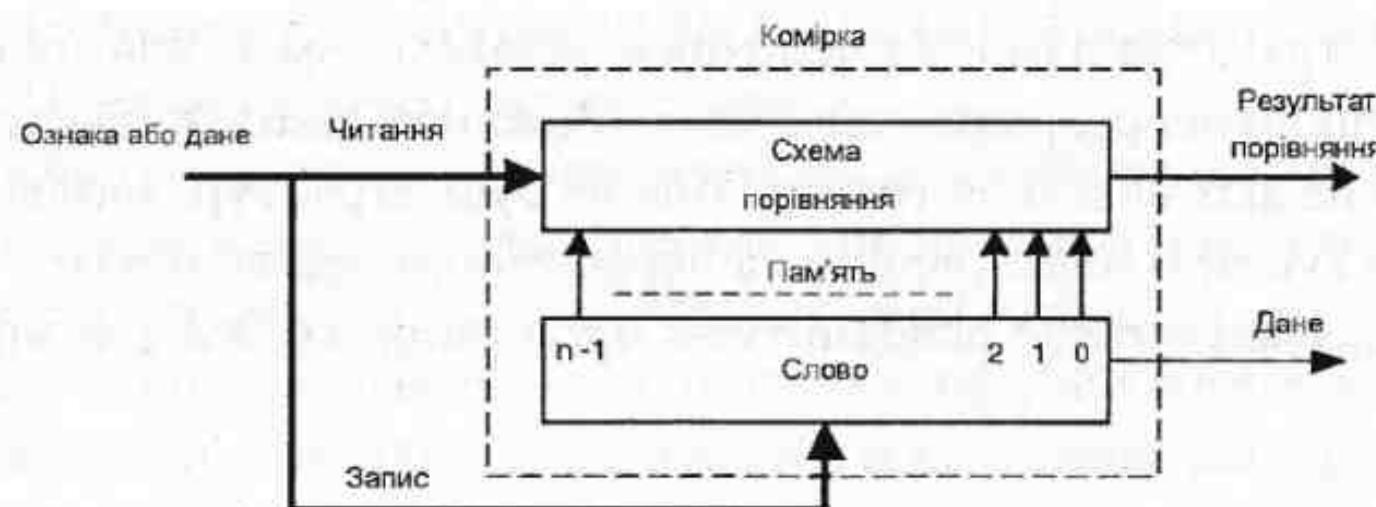


Рис. 9.17. Будова комірки пам'яті з повним паралельним асоціативним доступом

9.3.3. Пам'ять з неповним паралельним асоціативним доступом

Другий тип організації пам'яті з асоціативним доступом – пам'ять з неповним паралельним асоціативним доступом. В цій пам'яті проводиться послідовна обробка розрядів слів. Тут логічні операції порівняння визначені тільки для одного або декількох розрядів ознак даних, а не для всіх розрядів одночасно, як це є в пам'яті з повним паралельним асоціативним доступом. Наприклад, якщо порівняння визначені тільки для одного розряду ознак даних, і потрібно здійснити пошук даних, ознаки яких займають розряди 0–20, то спочатку потрібно сформувати команду для одночасного порівняння лише одного 0-го біту в кожному записі, потім 1-го, і т. д. Пам'ять даного типу має подібну загальну структуру до попередньої з тією відмінністю, що операції в пам'яті здійснюються у вигляді циклів операцій над окремими розрядами і тривалість операцій пропорційна довжині поля ознак, яке аналізується. Комірка даної пам'яті зображена на рис. 9.18.



Рис. 9.18. Комірка пам'яті з неповним паралельним асоціативним доступом з послідовною обробкою розрядів слів

Прикладом пам'яті з неповним паралельним асоціативним доступом, в якій здійснюється послідовна обробка розрядів, може бути пам'ять системи STARAN.

9.3.4. Пам'ять з послідовним асоціативним доступом

Третій тип організації пам'яті з асоціативним доступом – пам'ять з послідовним асоціативним доступом. Тут логічні операції порівняння виконуються лише над одним словом. Доступ до даних при цьому здійснюється послідовним застосуванням логічних операцій до окремих слів, які зберігаються в пам'яті. Загальна структура такої пам'яті зображена на рис. 9.19. Оскільки при послідовній обробці слів такі властивості асоціативної пам'яті як одночасність операцій та майже не залежна від числа регістрів пам'яті тривалість пошуку відсутні, така пам'ять не в повній мірі відповідає поняттю пам'яті з асоціативним доступом.

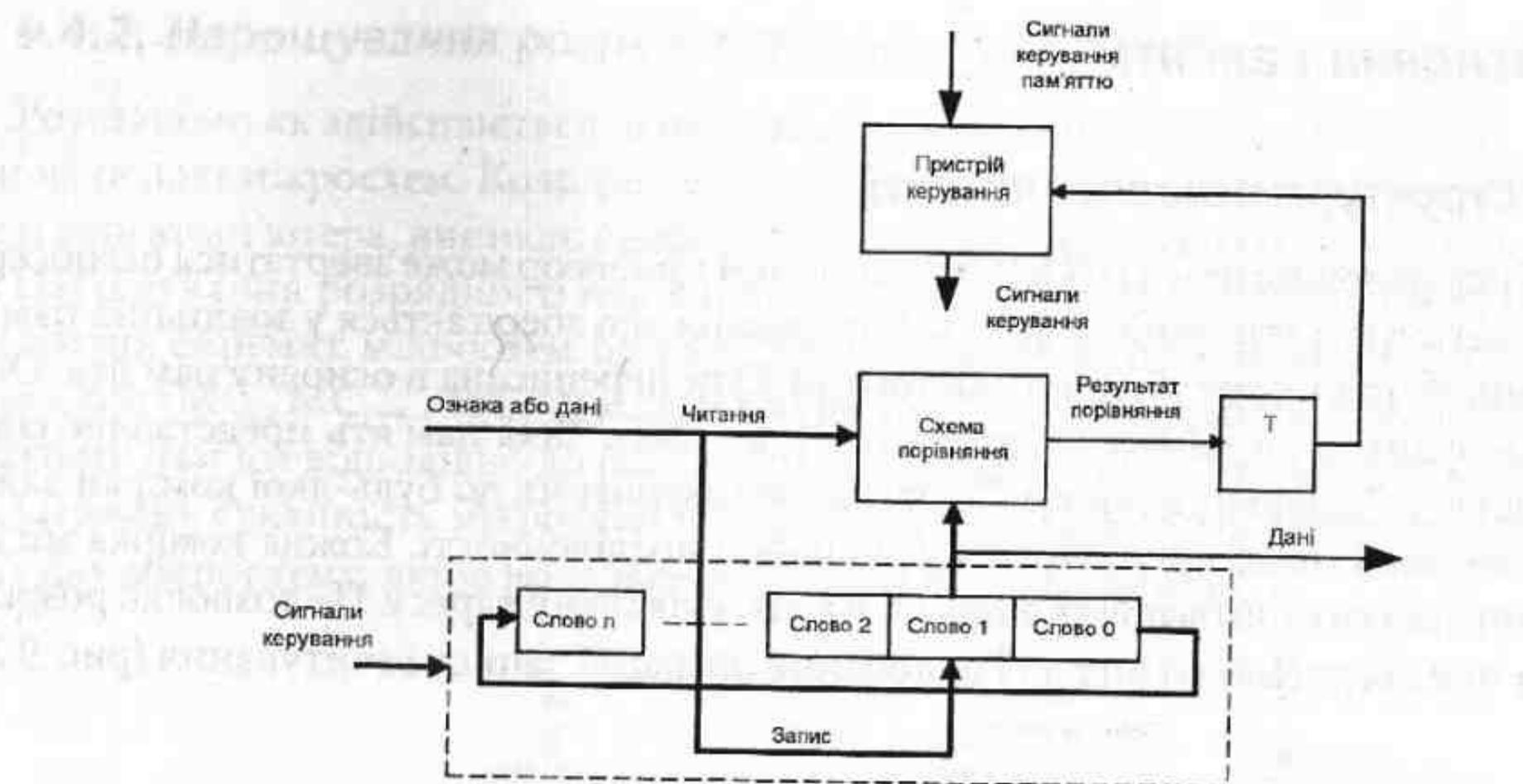


Рис. 9.19. Загальна структура пам'яті з послідовним асоціативним доступом

9.3.5. Пам'ять з частково асоціативним доступом

Четвертий тип організації пам'яті з асоціативним доступом – пам'ять із частково асоціативним доступом. Всі слова цієї пам'яті формуються в п груп. Кожна група слів обробляється паралельно з усіма іншими групами елементів шляхом послідовного переходу від одного слова групи до іншого. Отже, потрібні засоби, які дозволяють послідовно вибирати слова із груп для застосування до них заданих логічних операцій. Оскільки слова в групах вибираються послідовно, то можна обмежитись використанням пристрій пам'яті з послідовним доступом взамін більш дорогих пристрій пам'яті з асоціативним доступом. Загальна структура пам'яті з частково асоціативним доступом подана на рис. 9.20.

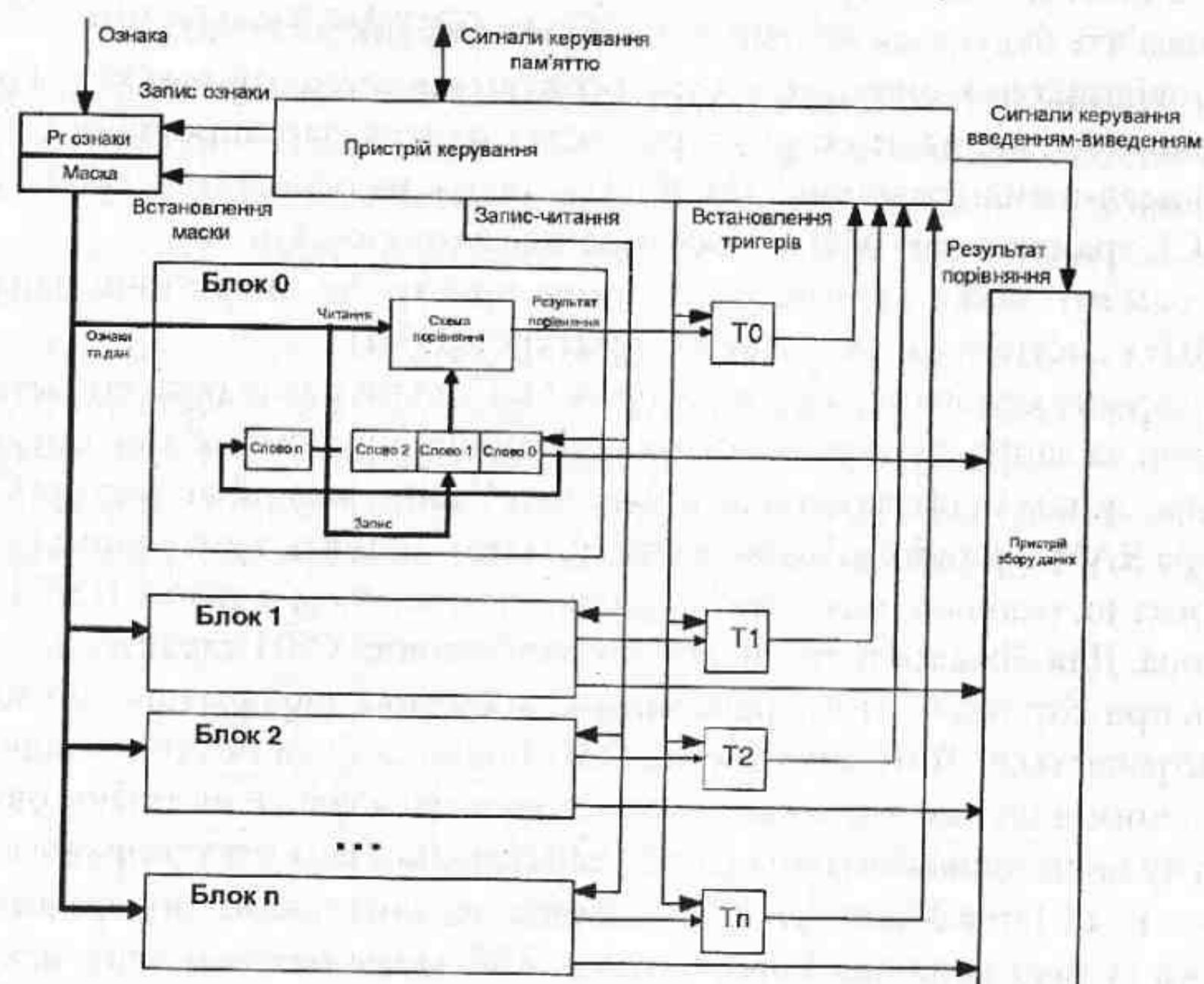


Рис. 9.20. Загальна структура пам'яті з частково асоціативним доступом

9.4. Основна пам'ять

9.4.1. Структура основної пам'яті

До комірок основної пам'яті (ОП) центральний процесор може звертатися безпосередньо, як і до регистрів регістрового файла. Інформація, що зберігається у зовнішній пам'яті, стає доступною процесору тільки після того, як буде переписана в основну пам'ять. Основну пам'ять будують як пам'ять з довільним доступом. Така пам'ять представляє собою масив комірок, а “довільний доступ” означає, що звернення до будь-якої комірки займає один і той же час і може проводитися у довільній послідовності. Кожна комірка містить фіксоване число запам'яточних елементів і має унікальну адресу. Це дозволяє розрізняти комірки при зверненні до них для виконання операцій запису і зчитування (рис. 9.21).



Рис. 9.21. Структура основної пам'яті

Раніше ОП комп'ютерів будувалась на феромагнітних кільцях. В сучасних комп'ютерах основна пам'ять будується на напівпровідниковых мікросхемах.

У напівпровідниковых мікросхемах застосовуються елементи пам'яті на основі: біполярних транзисторів; які мають структуру “метал-окисел-напівпровідник” (МОП); “метал-нітрид-окисел-напівпровідник” (МНОП); а також на основі пристрій із зарядовим зв'язком (ПЗС); транзисторів МОП з ізольованим затвором і ін.

Основна пам'ять може включати два типи пристрій: оперативні запам'яточуючі пристрій (ОЗП) і постійні запам'яточуючі пристрій (ПЗП).

Основну частину основної пам'яті утворює ОЗП, який називають оперативним, тому що він допускає як запис, так і зчитування інформації, причому обидві операції виконуються подібним чином та практично за той же час. У англомовній літературі ОЗП відповідає абревіатура RAM – Random Access Memory, тобто “пам'ять з довільним доступом”, що не зовсім коректно, оскільки пам'яттю з довільним доступом є також ПЗП і регістровий файл процесора. Для більшості типів напівпровідниковых ОЗП характерна енергозалежність – навіть при короткочасному перериванні живлення інформація, що зберігається в цій пам'яті, втрачається. Тому мікросхема ОЗП повинна бути постійно підключеною до джерела живлення, а ця пам'ять може використовуватися тільки як тимчасова пам'ять.

Другу групу напівпровідниковых пристрій основної пам'яті утворюють мікросхеми ПЗП (ROM — Read-Only Memory). ПЗП забезпечує зчитування інформації, але не допускає її зміни (у ряді випадків інформація в ПЗП може бути змінена, але цей процес суттєво відрізняється від зчитування і вимагає значно більшого часу).

9.4.2. Нарощування розрядності основної пам'яті

Розглянемо як здійснюється нарощування розрядності основної пам'яті при її побудові на основі мікросхем. Коли розрядність комірок у мікросхемі є меншою від розрядності слів комп'ютера, виникає необхідність об'єднання декількох мікросхем пам'яті.

Нарощування розрядності основної пам'яті реалізується шляхом об'єднання адресних входів окремих мікросхем пам'яті. Якщо об'єднати інформаційні входи і виходи п мікросхем ($MC_0, MC_1, \dots, MC_{n-1}$), кожна з яких має 2^m однорозрядних комірок, з входами і виходами пам'яті відповідно до рис. 9.22, то отримається пам'ять збільшеної розрядності. Одержану сукупність мікросхем називають модулем пам'яті. Модулем можна вважати і одну мікросхему, якщо вона вже має потрібну розрядність.

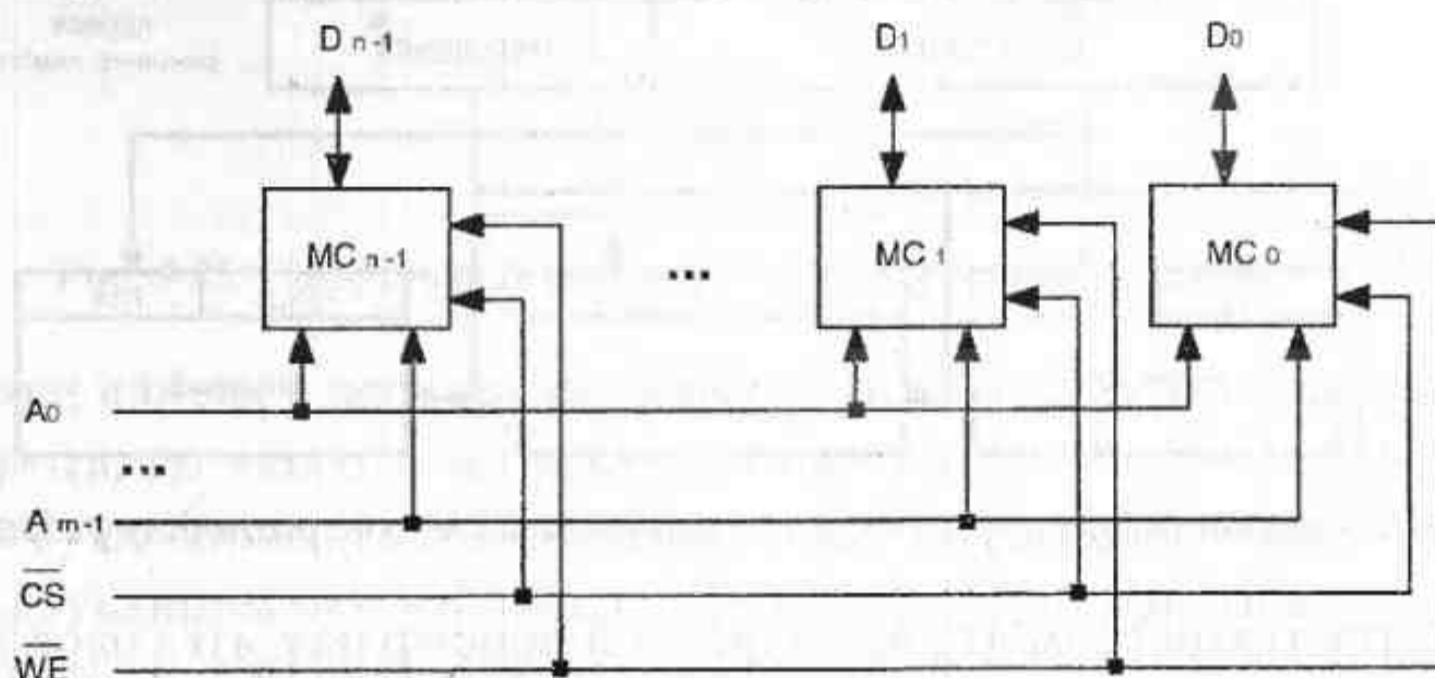


Рис. 9.22. Структура модуля пам'яті з n мікросхем

Тут позначення сигналів CS та WE означає відповідно вибірку кристала (Cristal Select) та дозвіл запису (Write Enable).

9.4.3. Нарощування ємності основної пам'яті

Ємність основної пам'яті сучасних комп'ютерів є настільки великою, що її не можна реалізувати на базі однієї мікросхеми. Для отримання необхідної ємності потрібно певним чином об'єднати декілька модулів пам'яті меншої ємності. У загальному випадку основна пам'ять комп'ютера практично завжди має блокову структуру, тобто містить декілька модулів.

При використанні блокової структури пам'яті, що складається з S модулів, адреса комірки перетвориться в пару (r, g) , де r – номер модуля, g – адреса комірки всередині модуля. Відомі три схеми розподілу розрядів адреси A між r та g .

В блоковій схемі розподілу розрядів адреси (рис. 9.23) номер модуля r визначають старші к розрядів адреси, які поступають на входи вибору модуля CS (Chip Select).

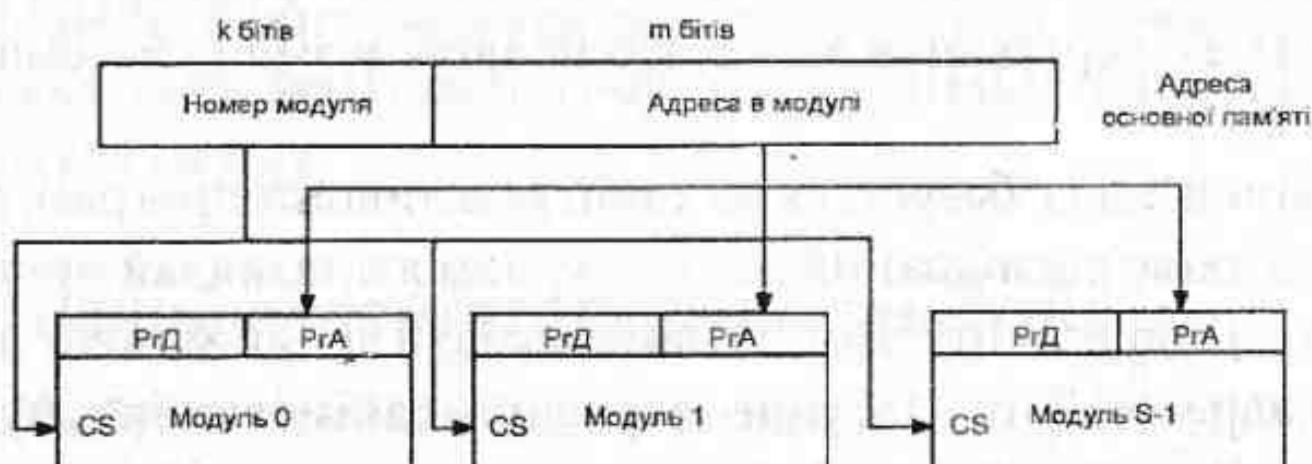


Рис. 9.23. Блокова структура пам'яті з блоковою схемою розподілу розрядів адреси

Адресний простір пам'яті розбитий на групи послідовних адрес, і кожна така група забезпечується окремим модулем пам'яті. Для звернення до основної пам'яті використовується n -розрядна адреса ($n = m + k$), m молодших розрядів якої ($A_{m-1} - A_0$) поступають паралельно до регістра адреси РгА всіх модулів пам'яті (МП) і вибирають в кожному з них одну комірку. Старші k розрядів адреси ($A_{n-1} - A_m$) містять номер модуля. У функціональному відношенні така пам'ять може розглядатися як єдина, ємність якої рівна сумарній ємності складових, а швидкодія – швидкодії окремого модуля.

В циклічній схемі розподілу розрядів адреси (рис. 9.24) номер p ($p = 0, 1, \dots, S-1$) модуля визначають k молодших розрядів адреси.

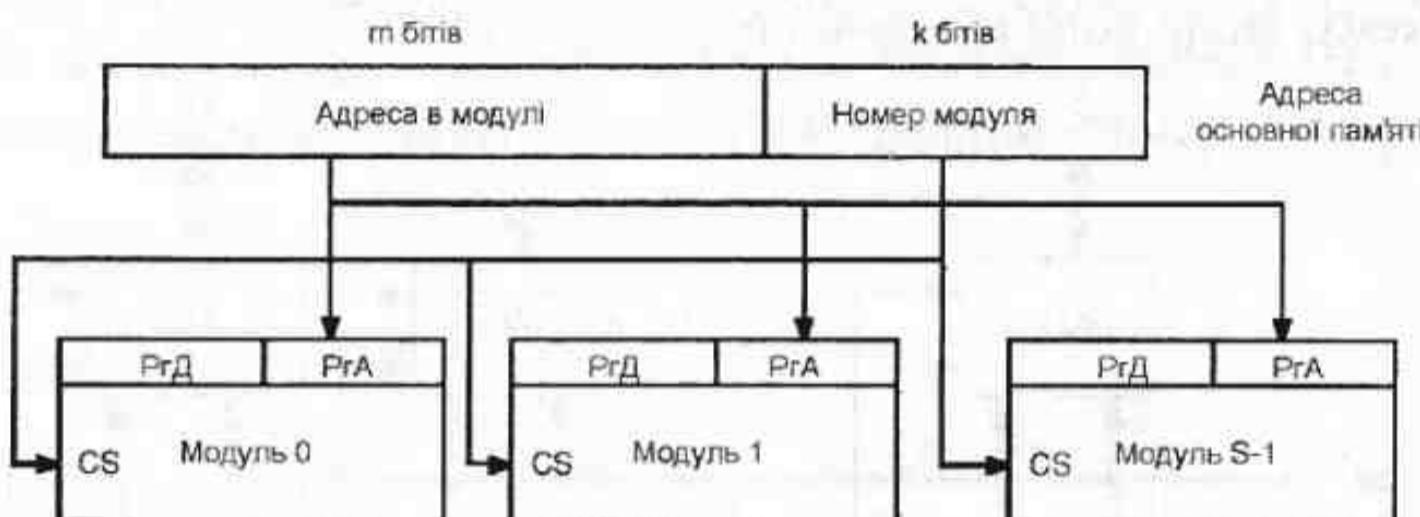


Рис. 9.24. Блокова структура пам'яті з циклічною схемою розподілу розрядів адреси

Адресний простір цієї пам'яті, як і адресний простір пам'яті з блоковою схемою розподілу розрядів адреси, розбитий на групи послідовних адрес, і кожна така група забезпечується окремим модулем пам'яті. Для звернення до основної пам'яті використовується n -розрядна адреса ($n = m + k$), m старших розрядів якої ($A_{n-1} - A_k$) поступають паралельно до регістра адреси РгА всіх модулів пам'яті (МП) і вибирають в кожному з них одну комірку. Молодші k розрядів адреси ($A_{k-1} - A_0$) містять номер модуля. У функціональному відношенні така пам'ять може розглядатися як єдина, ємність якої рівна сумарній ємності складових, а швидкодія – швидкодії окремого модуля.

Блоково-циклічна схема розподілу розрядів адреси є комбінацією двох попередніх схем.

9.4.4. Розшарування пам'яті

Крім придатності до нарощування ємності, блокова побудова пам'яті має іще одну перевагу – дозволяє скоротити час доступу до інформації. Це можливо завдяки потенційному паралелізму, властивому блоковій організації, який дозволяє досягти меншого часу доступу до інформації за рахунок паралельної роботи багатьох модулів пам'яті. Одну з використовуваних для цього методик називають розшаруванням пам'яті. У її основі лежить так зване чергування адрес, що полягає в зміні системи розподілу адрес між модулями пам'яті.

Прийом чергування адрес базується на такій властивості програм, як локальність за зверненням, згідно з якою послідовний доступ до пам'яті зазвичай проводиться до комірок, що мають суміжні адреси. Іншими словами, якщо в даний момент виконується звернення до комірки з адресою 7, то наступне звернення, найімовірніше, буде здійснюватись до комірки з адресою 8, пізніше 9 і т. д. Структура пристрою пам'яті, в якому використано чергування адрес з метою прискорення доступу до даних, наведена на рис. 9.25.

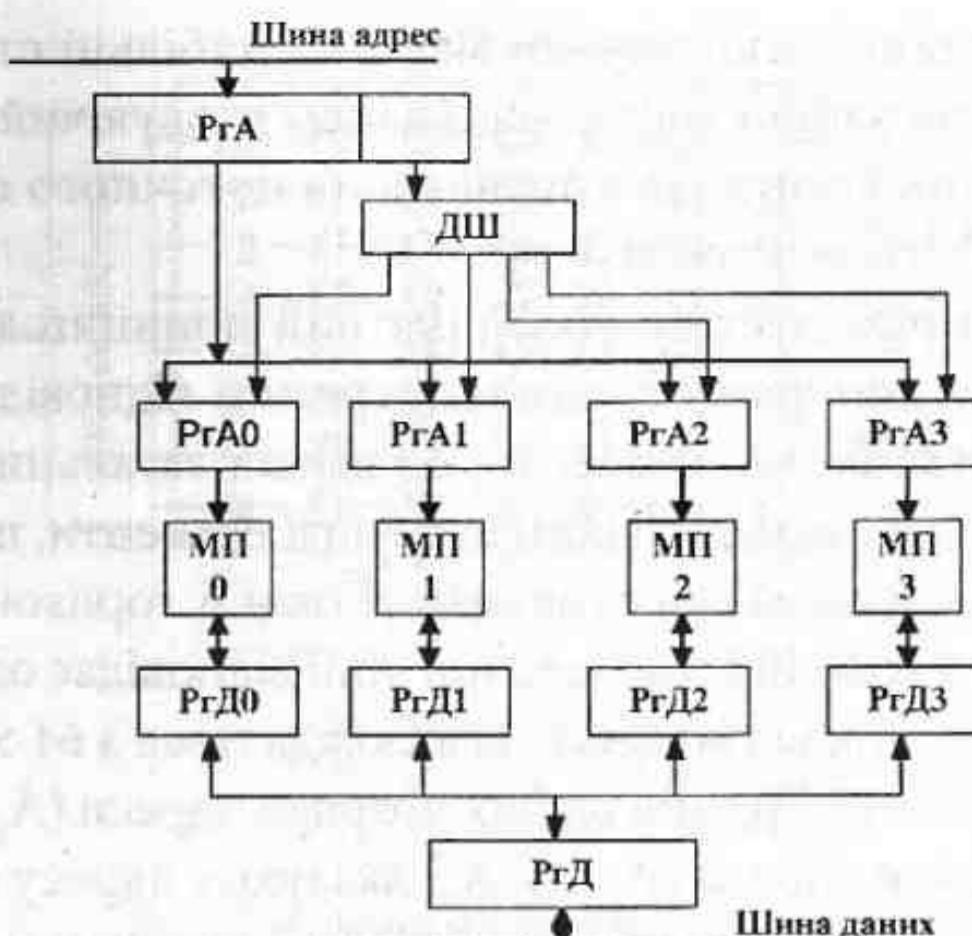


Рис. 9.25. Структура пристрою з розшаруванням пам'яті

Тут два розряди адреси з регістра адреси РгА поступають на дешифратор ДШ, виходи якого вказують, до якого з чотирьох регістрів адрес РгA0-РгA3 модулів пам'яті МП0-МП3 мають бути записані всі інші розряди з регістра адреси РгА. Принцип роботи пристрою з розшаруванням пам'яті пояснює рис. 9.26. Як видно з рисунку, до кожного з чотирьох модулів послідовні дані записуються по черзі.

Адреса слова	Модуль 0	Адреса слова	Модуль 1	Адреса слова	Модуль 2	Адреса слова	Модуль 3
0		1		2		3	
4		5		6		7	
8		9		10		11	
12		13		14		15	

Рис. 9.26. Принцип роботи пристрою з розшаруванням пам'яті

Тим самим забезпечується паралельна робота модулів пам'яті МП0-МП3, що дозволяє в S разів, де S – кількість модулів пам'яті (в чотири рази для прикладу, приведеного на рис. 9.26) прискорити роботу пам'яті.

Традиційні способи розшарування пам'яті добре працюють у рамках одного завдання, для якого характерна властивість локальності.

У багатопроцесорних системах із загальною пам'яттю, де запити на доступ до пам'яті достатньо незалежні, не виключений інший підхід, який можна розглядати як розвиток ідеї розшарування пам'яті. Для цього в систему включають декілька контролерів пам'яті, що дозволяє окремим модулям працювати автономно. Ефективність даного прийому залежить від частоти незалежних звернень до різних модулів. Кращого результату можна чекати при великому числі модулів, що зменшує вірогідність послідовних звернень до одного і того ж модуля пам'яті.

9.5. Оперативний запам'ятовуючий пристрій

Основна пам'ять будується на основі інтегральних мікросхем. Мікросхеми пам'яті організовані у вигляді матриці комірок, кожна з яких має n запам'ятовуючих елементів, де n – розрядність комірки, і має свою адресу. Кожен запам'ятовуючий елемент здатний

зберігати один біт інформації, оскільки він має два стабільні стани, які представляють двійкові значення 0 і 1. При запису інформації запам'ятовуючий елемент встановлюється в один із двох можливих станів. Для визначення поточного стану запам'ятовуючого елемента його вміст має бути зчитаний.

В мікросхемах пам'яті реалізується координатний принцип адресації комірок, згідно з яким комірка із заданим номером лежить на перетині відповідних вертикальної та горизонтальної ліній. Запам'ятовуючі елементи, об'єднані загальним горизонтальним провідником, прийнято називати рядком. Запам'ятовуючі елементи, підключені до загального вертикального провідника, називають стовпцем. Кожній горизонтальній лінії відповідає один з кодів адреси рядка, а кожній вертикальній лінії відповідає один з кодів адреси стовпця. На рис. 9.27 приведено приклад матриці, яка складається з 64-х комірок пам'яті з координатним принципом адресації. Три молодших розряди адреси (A_2, A_1, A_0) вказують адресу рядка, а три старших розряди адреси (A_5, A_4, A_3) вказують адресу стовпця. Так, комірка 27 лежить на перетині горизонтальної лінії з кодом 011 та вертикальної лінії з кодом 011.

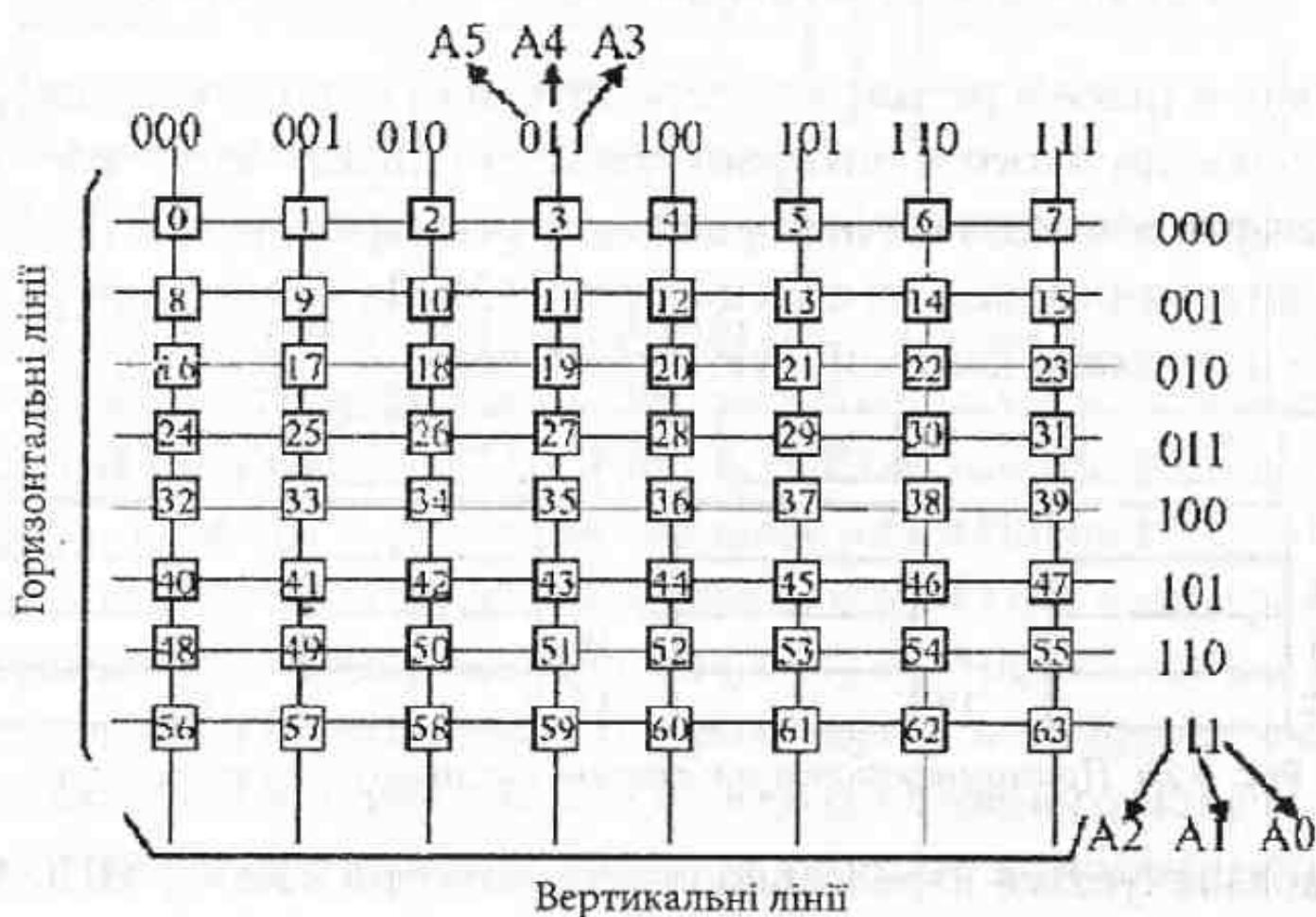


Рис. 9.27. Матриця комірок пам'яті з координатним принципом адресації

Адреса комірки, що поступає по шині адреси в мікросхему пам'яті, пропускається через логіку вибору, де вона розділяється на дві складові: адресу рядка і адресу стовпця. Адреси рядка і стовпця запам'ятаються відповідно в регістрі адреси рядка і регістрі адреси стовпця мікросхеми (рис. 9.28). Для зменшення числа контактів мікросхеми адреси рядка і стовпця в більшості мікросхем подаються в мікросхему через одні і ті ж контакти послідовно в часі (мультиплексуються). Кожен регістр з'єднаний зі своїм дешифраторм. Виходи дешифраторів утворюють систему горизонтальних і вертикальних провідників, до яких підключені матриці комірок пам'яті, при цьому кожна комірка пам'яті розташована на перетині одного горизонтального й одного вертикального провідників.

Крім адресних вертикальних провідників у мікросхемі повинна бути така ж кількість інформаційних провідників, по яких передаватиметься інформація, яка зчитується та записується до пам'яті. Сумісність запам'ятовуючих елементів і логічних схем, пов'язаних із вибором рядків і стовпців, називають ядром мікросхеми пам'яті (рис. 9.28).

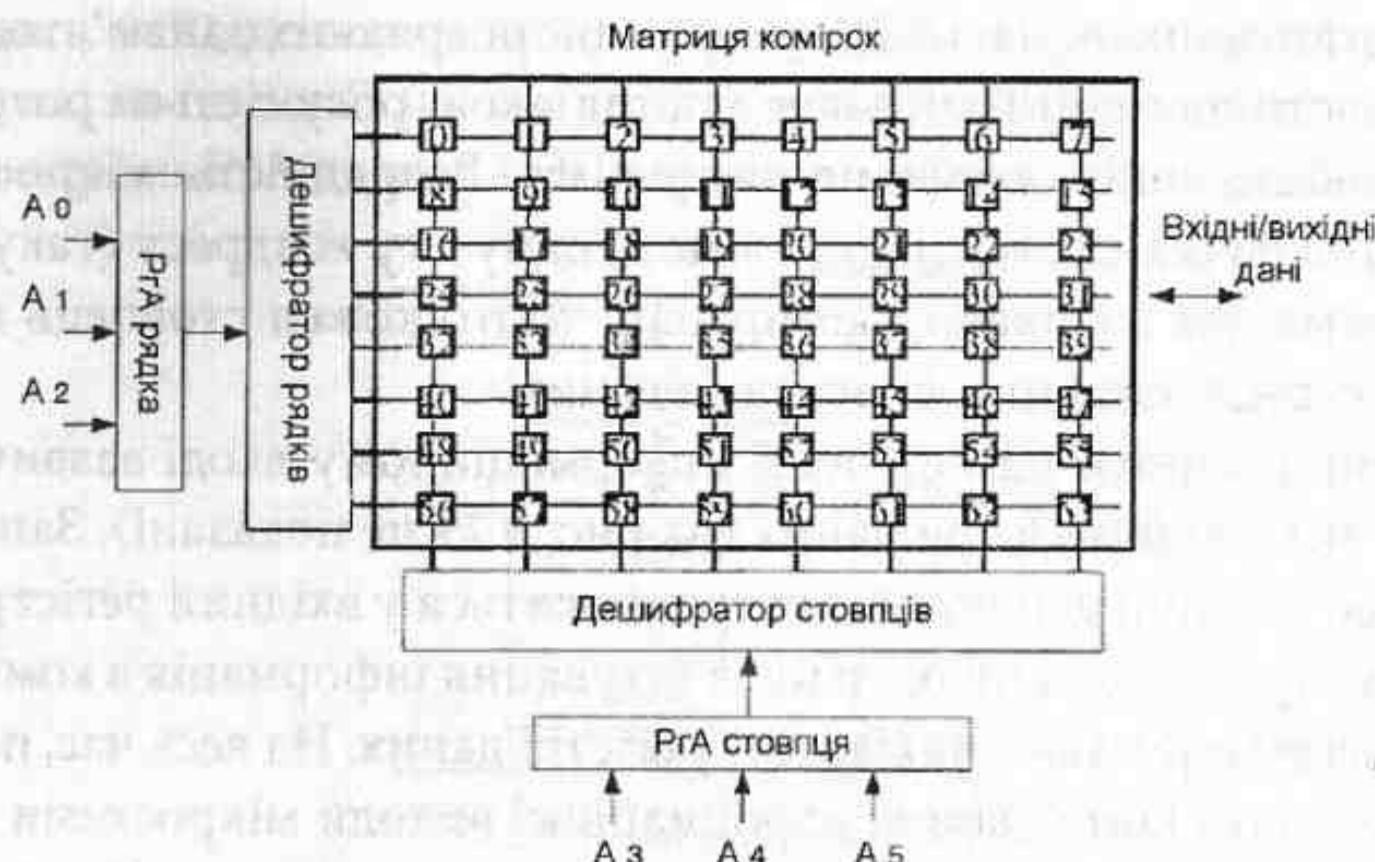


Рис. 9.28. Ядро мікросхеми пам'яті

Крім ядра, в мікросхемі є ще інтерфейсна логіка, що забезпечує взаємодію ядра із зовнішнім світом. У її завдання, зокрема, входить проведення комутації потрібного стовпця на вихід при читанні і на вхід при записі (рис. 9.29), яка здійснюється через вихідні ключі, що керуються логічними схемами запису і зчитування. При цьому логічні схеми запису і зчитування (логіка запису та логіка зчитування), а також логіка керування, яка задає режими роботи пам'яті, працюють на основі аналізу зовнішніх сигналів керування пам'яттю /RAS, /CE, /CS, /WE, /CAS.

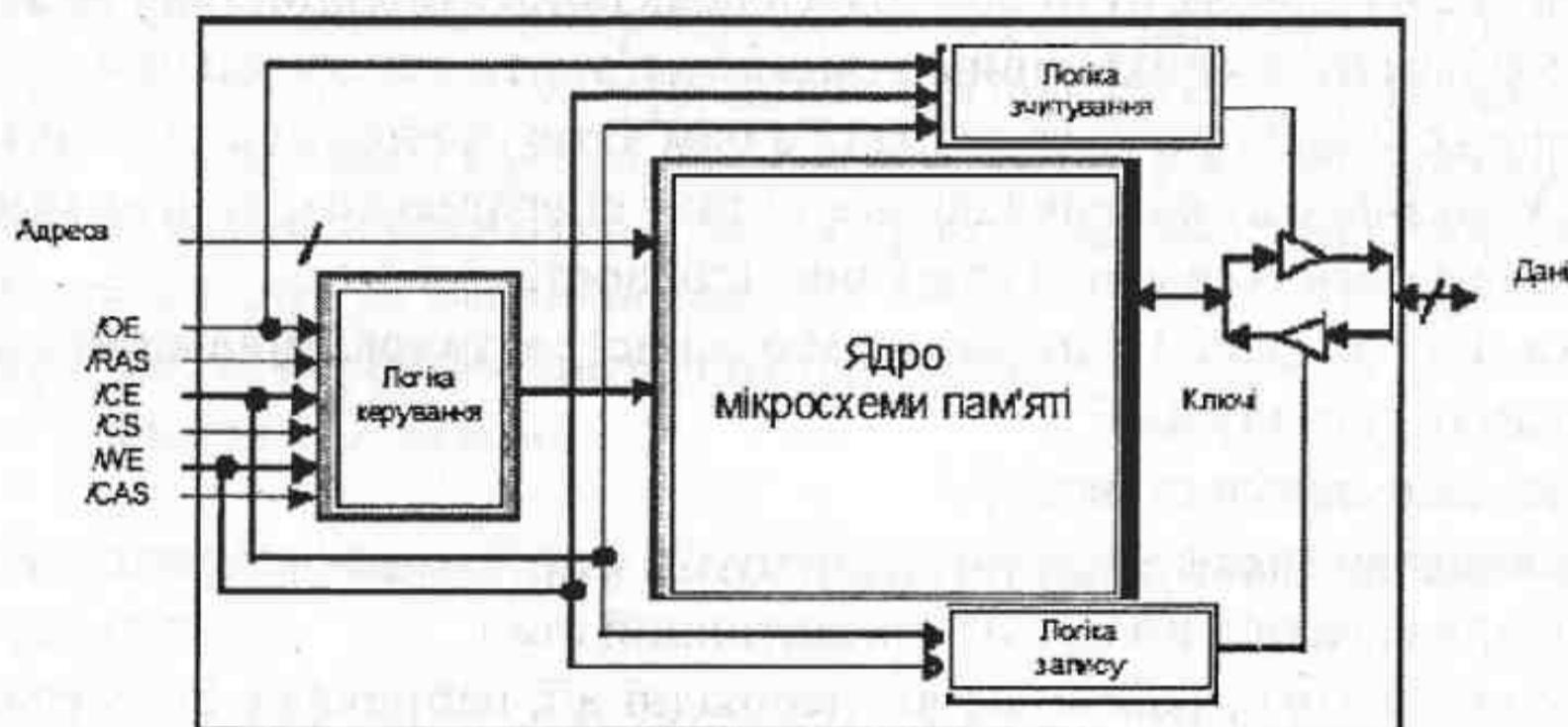


Рис. 9.29. Інтерфейсна логіка мікросхеми пам'яті

Для синхронізації процесів фіксації та обробки адресної інформації всередині мікросхеми адреса рядка (RA) супроводжується сигналом RAS (Row Address Strobe – строб рядка), а адреса стовпця (CA) – сигналом CAS (Column Address Strobe – строб стовпця). Щоб стробування було надійним, ці сигнали подаються із затримкою, достатньою для завершення переходних процесів на шині адреси та в адресних лініях мікросхеми.

Сигнал вибору мікросхеми CS (Chip Select) дозволяє роботу мікросхеми і використовується для вибору певної мікросхеми в системах пам'яті, що складаються з декількох мікросхем.

Сигнал WE (Write Enable – дозвіл запису) визначає вид виконуваної операції (зчитування або запис).

На фізичну організацію ядра, як матрицю однорозрядних запам'ятовуючих елементів, накладається логічна організація пам'яті, під якою розуміється розрядність мікросхеми, тобто кількість ліній введення-виведення. Розрядність мікросхеми визначає кількість запам'ятовуючих елементів, що мають одну і ту ж адресу (таку сукупність запам'ятовуючих елементів називають коміркою), тобто кожен стовпець містить стільки розрядів, скільки є ліній введення-виведення даних.

Для прискорення роботи пам'яті на її інформаційному вході зазвичай встановлюються вхідний та вихідний регістри даних (на рис. 9.29 не показані). Записувана інформація, що поступає по шині даних, спочатку заноситься у вхідний регістр даних, а потім у вибрану комірку. При виконанні операції зчитування інформація з комірки до її видачі на шину даних буферизується у вихідному регістрі даних. На весь час, поки мікросхема пам'яті не використовує шину даних, інформаційні виходи мікросхеми переводяться в третій (високоімпедансний) стан. Керування перемиканням в третій стан забезпечується сигналом OE (Output Enable – дозвіл видачі вихідних сигналів). Цей сигнал активізується при виконанні операції зчитування.

Для більшості перерахованих вище сигналів керування активним зазвичай вважається їх низький рівень, що і показано на рис. 9.29.

Керування операціями з основною пам'яттю здійснюється контролером пам'яті (рис. 9.21). Зазвичай цей контролер входить до складу центрального процесора або реалізується у вигляді зовнішнього по відношенню до пам'яті пристрою. В останніх типах мікросхем пам'яті частина функцій контролера покладається на мікросхему пам'яті. Хоча робота мікросхеми пам'яті може бути організована як по синхронній, так і по асинхронній схемі, контролер пам'яті є синхронним пристроєм, тобто він спрацьовує виключно по тактових імпульсах. З цієї причини операції з пам'яттю прийнято описувати з прив'язкою до тактів. У загальному випадку на кожну таку операцію потрібно як мінімум п'ять тактів, які використовуються у наступній послідовності:

- Вказівка типу операції (зчитування або запис) і встановлення адреси рядка.
- Формування сигналу RAS.
- Встановлення адреси стовпця.
- Формування сигналу CAS.
- Повернення сигналів RAS і CAS в неактивний стан.

Даний перелік враховує далеко не всі необхідні дії, наприклад, регенерацію вмісту пам'яті в динамічних ОЗП.

9.6. Постійний запам'ятовуючий пристрій

9.6.1. Організація роботи постійного запам'ятовуючого пристрою

Постійний запам'ятовуючий пристрій (ПЗП), як і оперативний, є складовою частиною основної пам'яті, та часто використовується при побудові інших вузлів комп'ютера, наприклад, табличних операційних пристрій, мікропрограмних пристрій керування і т. д. Слово "постійний" вказує на те, що в цьому пристрії один раз записана інформація не міняється взагалі, або це здійснюється при переведенні його в спеціальний режим роботи. При цьому в ньому інформація зберігається і при відсутності напруги живлення.

Як видно з рис. 9.30, де показано інтерфейс ПЗП, в ньому відсутній сигнал запису-зчитування даних, як це є в ОЗП, і дане зчитується з ПЗП в регістр даних РгД при поданні з регістра адреси РгА на вхід ПЗП адреси відповідної комірки пам'яті.



Рис. 9.30. Інтерфейс ПЗП

Для підвищення швидкодії і збільшення об'єму ПЗП використовуються ті ж підходи, що і для ОЗП. Мікросхеми ПЗП, як і мікросхеми ОЗП, побудовані за принципом матричної структури накопичувача, де у вузлах розміщені запам'яточні елементи, підключенні до адресних та інформаційних ліній. В якості запам'яточних елементів тут можуть бути використані провідники, діоди або транзистори, які можуть бути замкнуті (відповідає значенню 1), або розімкнуті (відповідає значенню 0).

Основним режимом роботи ПЗП є зчитування інформації, яке мало відрізняється від аналогічної операції в ОЗП як за організацією, так і за тривалістю. Саме ця обставина підкреслює англомовну назву ПЗП – ROM (Read-Only Memory – пам'ять тільки для зчитування). В той же час запис в ПЗП у порівнянні зі зчитуванням зазвичай є складнішим і потребує великих витрат часу і енергії. Занесення інформації в ПЗП називають програмуванням або “прошивкою”. Сучасні ПЗП реалізуються у вигляді напівпровідниківих мікросхем, які за можливостями і способами програмування розділяють на:

- запрограмовані при виготовленні;
- одноразово програмовані після виготовлення;
- багаторазово програмовані.

9.6.2. Запрограмований при виготовленні постійний запам'яточний пристрій

Групу ПЗП, що програмуються при виготовленні, утворюють так звані масочні пристрії і саме до них прийнято застосовувати абревіатуру ПЗП. У літературі більш поширене позначення різних варіантів ПЗП скороченнями, ніж англійські назви; тому надалі також використовуватимемо аналогічну систему. Для маскованих ПЗП таким позначенням є ROM, співпадаючий із загальною назвою всіх типів ПЗП. Іноді такі мікросхеми іменують MROM (Mask Programmable ROM – ПЗП, що програмуються за допомогою маски).

Занесення інформації в масковані ПЗП складає частину виробничого процесу і полягає у підключені або не підключені запам'яточного елемента до розрядної лінії зчитування. Залежно від цього із запам'яточного елемента завжди зчитуватиметься 1 або 0. В ролі перемички виступає транзистор, розташований на перетині адресної і розрядної ліній. Які саме запам'яточні елементи повинні бути підключенні до вихідної лінії визначає маска, що дозволяє вибрати визначені ділянки кристала. При створенні маскованих ПЗП застосовуються різні технології. У першому випадку маска просто не

допускає металізації ділянки, яка з'єднує транзистор з розрядною лінією зчитування. Друга технологія пов'язана з типом транзистора у вузлі. Маска визначає, який польовий транзистор має бути імплантований в даний вузол, що працює в збагаченому режимі або в режимі збіднення. У третьому варіанті маска задає товщину оксидного шару транзистора. Залежно від цього на кристалі формується або стандартний транзистор, або транзистор з високим порогом спрацьування.

У початковий період масковані мікросхеми були дорогі, проте зараз це один з найбільш дешевих видів ПЗП. Для ПЗП характерна висока щільність пакування запам'ятовуючих елементів на кристалі і високі швидкості зчитування інформації. Основною сферою застосування є пристрой, що вимагають зберігання фіксованої інформації. Так, подібні ПЗП часто використовують для зберігання мікропрограм у мікропроцесорах, констант у табличних операційних пристроях, шрифтів у лазерних принтерах.

9.6.3. Одноразово запrogramований після виготовлення постійний запам'ятовуючий пристрій

Створення масок для ПЗП виправдане при виробництві великого числа копій. Якщо потрібна відносно невелика кількість мікросхем із даною інформацією, розумною альтернативою є одноразово програмовані ПЗП. Структура матриці вентилів програмованого ПЗП показана на рис. 9.31.

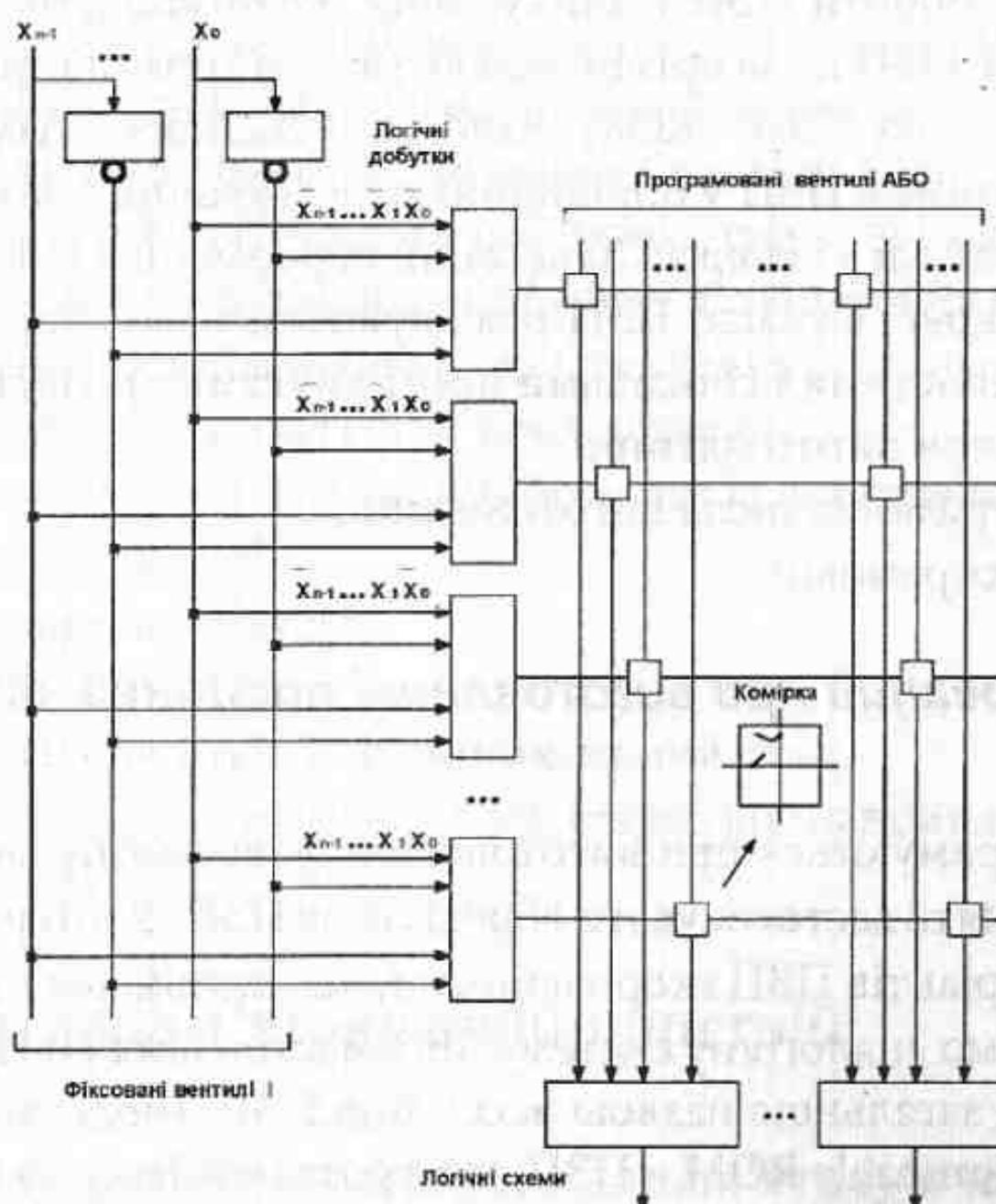


Рис. 9.31. Структура програмованого ПЗП

Одноразово програмований ПЗП складається з двох матриць І та АБО. Потрібно зauważити, що матриця І представляє собою декодер, який генерує всі логічні добутки п змінних. Зафарбовані чорним круги зображають фіксовані електричні з'єднання між входами ПЗП та комірками І. З цієї причини ця матриця називається фіксованою матрицею І.

Кожна функція формується однією коміркою АБО, яка має 2^n входів. З'єднання між виходами результуючих добутків і входами комірок АБО задаються користувачем шляхом програмування комірок, які відображають з'єднання добутку та комірки АБО. В противагу фіксованого масиву вентилів І масив вентилів АБО є програмовним. На рис. 9.31 показано збільшене зображення програмової комірки. Ця комірка включає ключ, який може бути відкритим або закритим. Закритий ключ означає включення добутку до результату, формуючи функцію 1 для вхідної комбінації. Коли ж ключ відкритий, добуток не включається і для цього входу функція рівна 0.

Оскільки ПЗП генерує всі можливі логічні добутки, будь-яка функція п змінних може бути реалізована.

Одноразово програмованих ПЗП є кілька типів

Мікросхеми типу PROM (Programmable ROM – програмовані ПЗП) інформація може бути записана тільки одноразово. Першими такими ПЗП стали мікросхеми пам'яті на базі плавких запобіжників. У початковій мікросхемі у всіх вузлах адресні лінії з'єднані з розрядними. Занесення інформації в PROM проводиться електричними сигналами шляхом перепалювання окремих плавких перемичок, і може бути виконано постачальником або споживачем через деякий час після виготовлення мікросхеми. Подібні ПЗП випускалися в рамках серій K556 і K1556. Пізніше з явилися мікросхеми, де в перемичку входили два діоди, включенні назустріч один одному. В процесі програмування можна було видалити перемичку за допомогою електричного пробою одного з цих діодів. У будь-якому варіанті для запису інформації потрібне спеціальне устаткування – програматори. Основними недоліками даного виду ПЗП були великий відсоток браку і необхідність спеціального термічного тренування після програмування, без якого надійність зберігання даних була невисокою.

Ще один вид одноразово програмованого ПЗП – це OTP EEPROM (One Time Programmable EEPROM – EEPROM з одноразовим програмуванням). У його основі лежить кристал EEPROM (див. нижче), але поміщений в дешевий непрозорий пластиковий корпус без кварцевого вікна, із за чого він може бути запрограмований лише один раз.

9.6.4. Багаторазово програмований постійний запам'ятовуючий пристрій

В багаторазово програмованих ПЗП в програмованих комірках використовуються МОП транзистори, параметри яких змінюються під впливом високої напруги. Вміст ПЗП може бути стертій застосуванням ще одної високої напруги або застосуванням ультрафіолетового світла.

Процедура програмування таких ПЗП складається з двох етапів. Спочатку проводиться стирання вмісту всіх або частини комірок, а потім проводиться запис нової інформації. У цьому класі ПЗП виділяють декілька груп:

- EPROM (Erasable Programmable ROM – програмовані ПЗП, вміст яких може бути стертій);
- EEPROM (Electrically Erasable Programmable ROM – програмовані ПЗП, вміст яких може бути стертій електрично);
- флеш пам'ять

У EPROM запис інформації проводиться електричними сигналами, так само як в PROM, проте перед операцією запису вміст всіх комірок повинен бути приведений до одинакового стану (стерто) шляхом дії на мікросхему ультрафіолетовим опромінюванням. Кристал по-

міщається в керамічний корпус, що має невелике кварцове вікно, через яке і проводиться опромінювання. Щоб запобігти випадковому стиранню інформації, після опромінювання кварцове вікно заклеють непрозорою плівкою. Процес стирання може виконуватися багато разів. Кожне стирання займає близько 20 хвилин. Цикл програмування займає декілька сотень мілісекунд. Час зчитування є близьким до показників ROM і DRAM.

У порівнянні з PROM мікросхеми EPROM є дорожчими, але завдяки можливості багатократного перепрограмування вони часто застосовуються. Даний вид мікросхем випускається в рамках серії K573.

Привабливішим варіантом багатократно програмованої пам'яті є програмована постійна пам'ять EEPROM, вміст якої може бути стертий електрично. Стирання і запис інформації в цю пам'ять проводяться побайтово, причому стирання – не окремий процес, а лише етап, що відбувається автоматично при записі. Операція запису займає істотно більше часу ніж зчитування – декілька сотень мікросекунд на байт. У мікросхемі використовується той же принцип зберігання інформації, що і в EPROM. Програмування EEPROM не вимагає спеціального програматора і реалізується засобами самої мікросхеми. Випускаються два варіанти мікросхем EEPROM: з послідовним і паралельним доступом, причому перших значно більше. У EEPROM з доступом по послідовному каналу (SEEPROM – Serial EEPROM) адреси, дані і команди передаються по одному провіднику і синхронізуються імпульсами на тактовому вході. Перевагою SEEPROM є малі габарити і мінімальне число ліній введення-виведення, а недоліком – великий час доступу. SEEPROM випускається в рамках серій мікросхем 24Cxx, 25Cxx і 28Cxx, а паралельні EEPROM – в складі серії 28Cxx.

В цілому EEPROM є дорожчими, ніж EPROM, а мікросхеми мають менш щільне пакування комірок, тобто меншу ємність.

Відносно новий вид напівпровідникової пам'яті – це флеш-пам'ять (назву flash можна перевести як “спалах блискавки”, що підкреслює відносно високу швидкість перепрограмування). Вперше анонсована у середині 80-х років, флеш-пам'ять багато в чому схожа на EEPROM, але використовує особливу технологію побудови запам'ятовуючих елементів. Analogічно EEPROM, у флеш-пам'яті стирання інформації проводиться електричними сигналами, але не побайтово, а по блоках або повністю. Тут слід зазначити, що існують мікросхеми флеш-пам'яті з розбиттям на дуже дрібні блоки (сторінки) з автоматичним посторінковим стиранням, що зближує їх за можливостями з EEPROM. Як і у випадку з EEPROM, мікросхеми флеш-пам'яті випускаються у варіантах з послідовним і паралельним доступом.

За організацією масиву запам'ятовуючих елементів розрізняють флеш-пам'ять з тотальним очищеннем, коли стирання допустимо тільки для всього масиву запам'ятовуючих елементів, та з блоковим очищеннем, коли масив запам'ятовуючих елементів роздільний на декілька блоків різного або однакового розміру, вміст яких може очищатися незалежно.

Вміст флеш-пам'яті може бути очищений за декілька секунд, запис одного слова займає декілька мікросекунд, а час доступу до даних при зчитуванні – декілька десятків наносекунд.

9.7. Зовнішня пам'ять

9.7.1. Магнітні диски

Магнітні диски – це набір металевих або пластмасових пластин, покритих магнітним матеріалом, на якому зберігається інформація. Запис та зчитування інформації проводиться за допомогою головки запису-зчитування, яка є електромагнітною котушкою, причому в процесі запису-зчитування диск обертається відносно нерухомої головки.

При записі на головку подаються електричні імпульси, що намагнічують ділянку поверхні магнітного матеріалу під нею, причому характер намагніченості поверхні різний залежно від напряму струму в котушці. Зчитування базується на електричному струмі, що наводиться в котушці головки, коли під нею проходить ділянка магнітного матеріалу поверхні диска, причому в котушці наводиться струм тієї ж полярності, що використовувався для запису інформації. Не дивлячись на різноманітність типів магнітних дисків, принципи їх організації зазвичай однотипні.

Дані на диску організовані у вигляді набору концентричних кіл, названих доріжками (рис. 9.26). Кожна з них має ту ж ширину, що і головка. Сусідні доріжки розділені проміжками. Це запобігає помилкам із-за зсуву головки або із-за інтерференції магнітних полів. Як правило, для спрощення електроніки приймається, що на всіх доріжках може зберігатися однаакова кількість інформації. Таким чином, щільність запису збільшується від зовнішніх доріжок до внутрішніх.

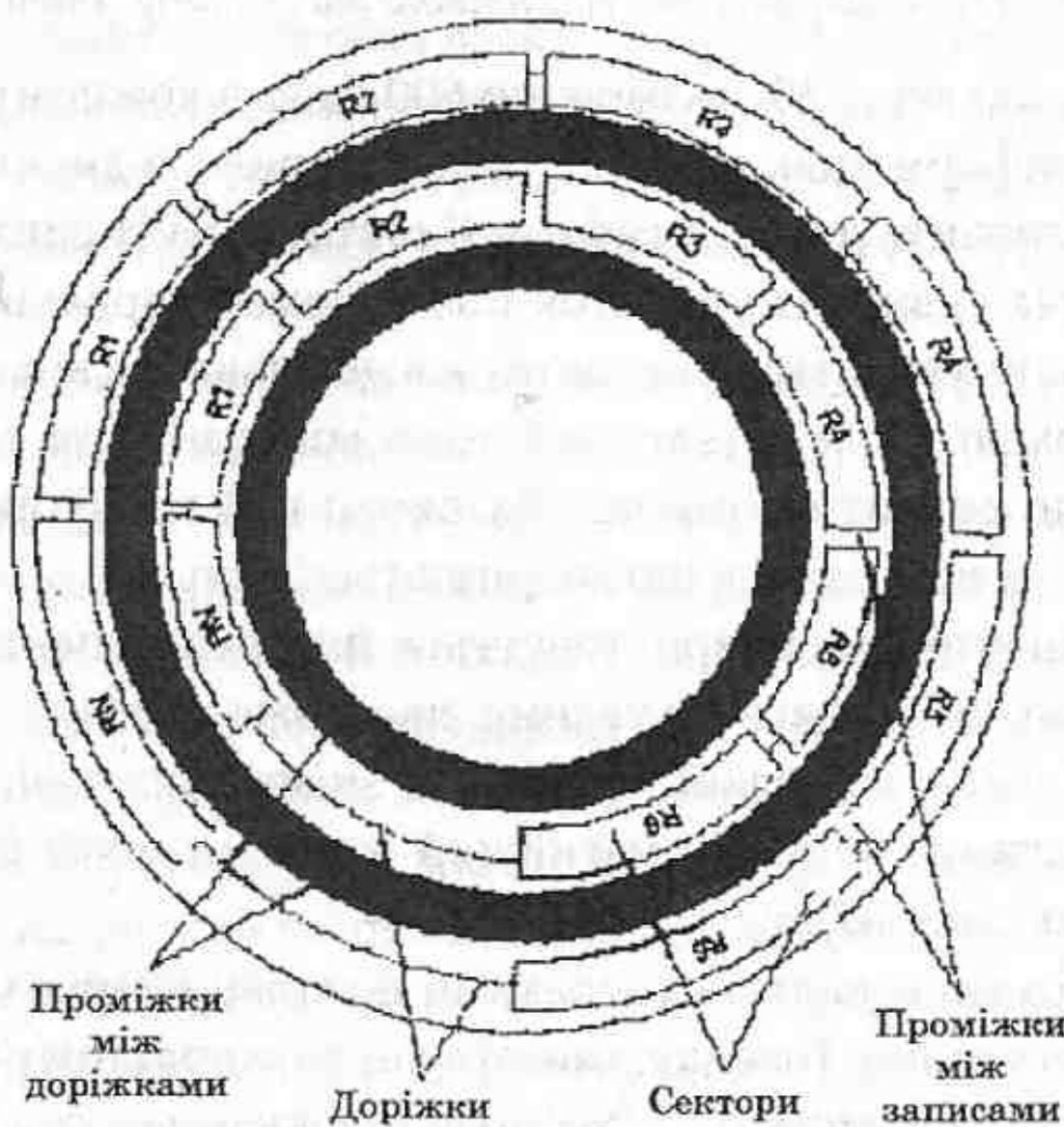


Рис. 9.32. Розміщення інформації на магнітному диску

Обмін інформацією між ОП і магнітним диском здійснюється блоками. Розмір блоку зазвичай є меншим ємності доріжки, і дані на доріжці зберігаються у вигляді послідовних областей – секторів, розділених між собою проміжками. Розмір сектора рівний мінімальному розміру блоку.

Типове число секторів на доріжці коливається від 10 до 100. При такій організації повинні бути задані точка відліку секторів і спосіб визначення початку і кінця кожного сектора. Все це забезпечується за допомогою форматування, в ході якого на диск заноситься службова інформація, недоступна користувачу і використовувана тільки апаратурою дискової пам'яті.

Приклад розмітки магнітного диску показаний на рис. 9.33.

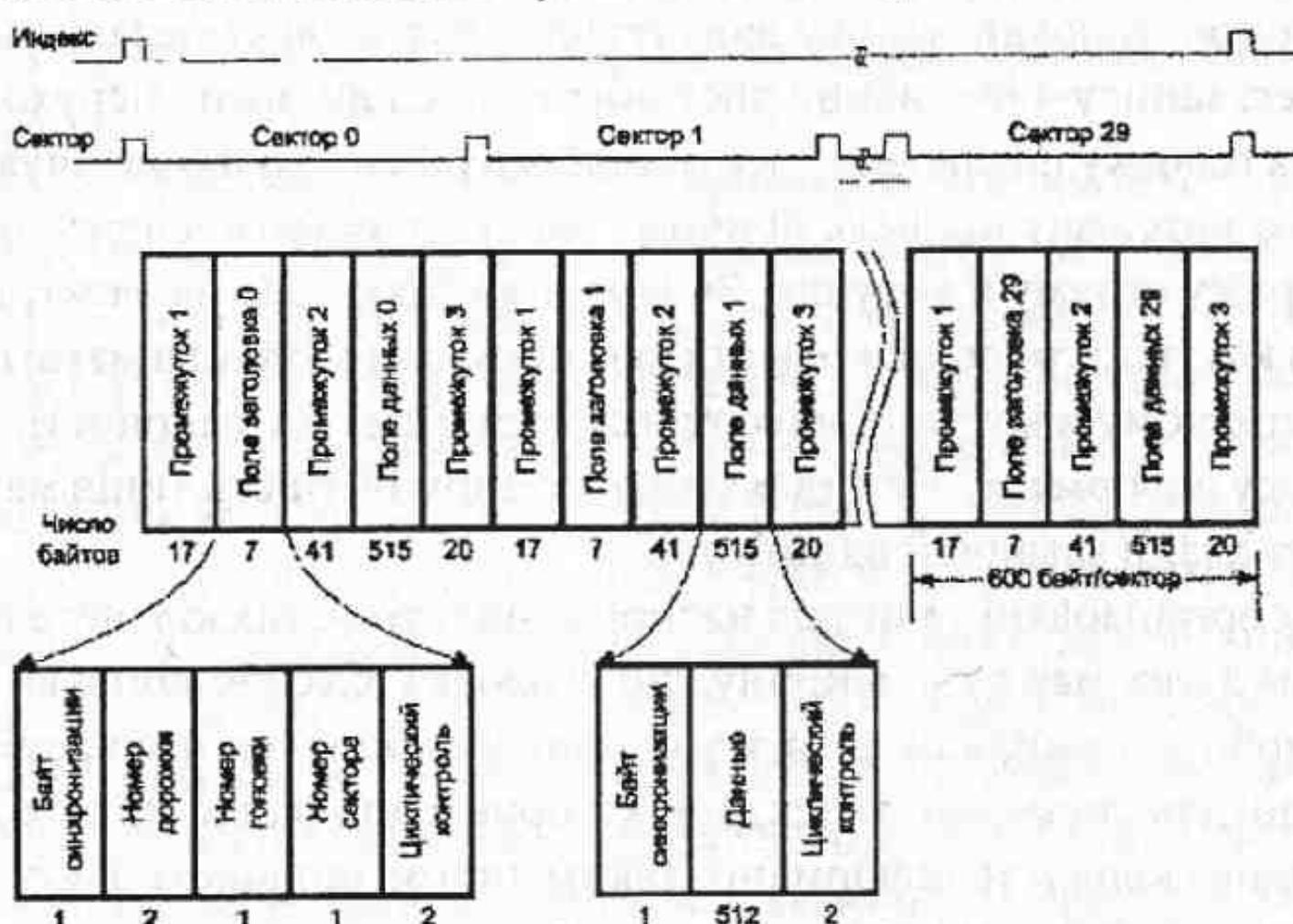


Рис. 9.33. Приклад розмітки магнітного диску типу "Вінчестер"

Тут кожна доріжка включає 30 секторів по 600 байт в кожному. Сектор зберігає 512 байт даних і керуючу інформацію, потрібну для контролера диска. Поле заголовка містить інформацію, що слугує для ідентифікації сектора. Байт синхронізації є характерним кодом, що дозволяє визначити початок поля. Номер доріжки визначає доріжку на поверхні. Якщо в накопичувачі використовується декілька дисків, то номер головки визначає потрібну поверхню. Поле заголовка і поле даних містять також код циклічного контролю, що дозволяє виявити помилки. Зазвичай цей код формується послідовним додаванням за модулем 2 всіх байтів, що зберігаються в полі.

В даний час у комп'ютерах використовується широкий спектр дискових систем: з фіксованою та рухомою головкою зчитування, яка може контактувати з магнітним матеріалом, або бути від нього на деякій відстані, із знімними та фікованими дисками, з одним та багатьма дисками, причому магнітний матеріал може бути нанесеним як на одну сторону диска, так і на обидві.

У дисковій системі з фіксованими головками на кожну доріжку приходиться по одній головці запису-зчитування. Головки змонтовані на жорсткому важелі, що перетинає всі доріжки диска. У дисковій системі з рухомими головками є тільки одна головка, також встановлена на важелі, проте важіль здатний переміщатися в радіальному напрямі над поверхнею диска, забезпечуючи можливість позиціювання головки на будь-яку доріжку.

Диски з магнітним носієм встановлюються в дискову систему, що складається з важеля, шпинделя, що обертає диск, і електронних схем, потрібних для введення і виведення двійкових даних. Незнімний диск зафікований в дисковій системі. Знімний диск може бути вийнятий і замінений на інший аналогічний диск. Перевага системи із знімними

дисками – можливість зберігання необмеженого об'єму даних при обмеженому числі дискових пристрій. Крім того, такий диск може бути перенесений з одного комп'ютера на інший.

Більшість дисків мають магнітне покриття з обох сторін.

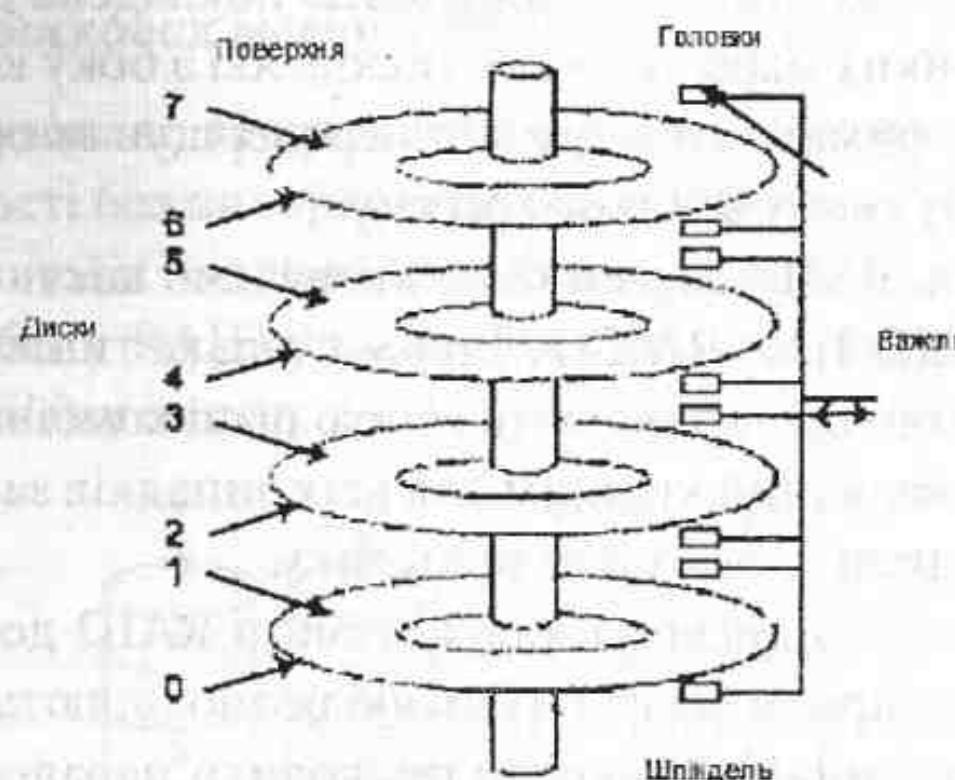


Рис. 9.34. Дискова пам'ять з дисковим пакетом

На осі може розташовуватися один, або декілька (рис. 9.34) дисків. У останньому випадку використовують термін дисковий пакет.

Залежно від вживаної головки запису-читування можна виділити три типи дискової пам'яті. У першому варіанті головка встановлюється на фіксованій дистанції від поверхні так, щоб між ними залишався повітряний проміжок. Другий варіант – це коли в процесі зчитування і запису головка і диск знаходяться у фізичному контакті. Такий спосіб використовується, наприклад, у накопичувачах на гнучких магнітних дисках (дискетах).

Для правильного запису і зчитування головка повинна генерувати і сприймати магнітне поле певної інтенсивності, тому чим вужча головка, тим біжче повинна вона розміщуватися до поверхні диска (вужча головка означає і вужчі доріжки, а значить, і велику місткість диска). Проте наближення головки до диска означає і більший ризик помилки за рахунок забруднення і дефектів. В процесі вирішення цієї проблеми був створений диск типу вінчестер. Головки вінчестера і диски поміщені в герметичний корпус, що захищає їх від забруднення. Крім того, головки мають дуже малу вагу й аеродинамічну форму. Вони спроектовані для роботи значно біжче до поверхні диска, ніж головки в звичайних жорстких дисках, тим самим підвищується щільність зберігання даних. При зупиненному диску головка прилягає до його поверхні. Тиску, що виникає при обертанні диска, достатньо для підйому головки над поверхнею. В результаті створюється неконтактна система з вузькими головками запису-читування, що забезпечує щільніше прилягання головки до поверхні диска, ніж у звичайних жорстких дисках.

9.7.2. Масиви магнітних дисків з надлишковістю

Оскільки жорсткі диски є основною зовнішньою пам'яттю комп'ютера, до їх характеристик, а саме вартості, продуктивності та відмовостійкості, висуваються підвищені вимоги. Причому, враховуючи те, що розрив у продуктивності між жорстким диском

і ядром комп'ютера постійно росте, а збільшення його продуктивності вдвічі було досягнуто лише через 10 років (рис. 9.8), розробники вимушенні шукати структурні методи покращення їх характеристик. Один із таких методів, а саме надлишкові масиви дешевих дисків (RAID – Redundant Arrays of Inexpensive Disks) був запропонований у 1987 році Д. Паттерсоном, Р. Гартом Р. Катцом. В основу RAID покладена наступна ідея: об'єднуючи в масив декілька невеликих і/або дешевих дисків, які з боку комп'ютера сприймаються як один диск, можна отримати систему, що перевершує за об'ємом, продуктивністю роботи і надійністю дорогі диски великого об'єму.

Асоціацією виробників RAID-систем було визначено шість базових типів дискових масивів RAID: RAID 0, RAID 1, ..., RAID 5. Також є кілька типів RAID, які не включені в даний перелік, оскільки вони представляють собою різні комбінації базових рівнів.Хоча ні один тип не може бути визнаний кращим для всіх випадків застосування, кожен з них дозволяє відчутно покращити якусь характеристику.

Підвищена продуктивність дискової підсистеми в RAID досягається за допомогою розбиття даних і дискового простору на смуги, які розподіляються по різних дисках, відповідно до визначеній системи. Це дозволяє проводити паралельне зчитування або запис відразу кількох смуг, якщо вони розміщені на різних дисках. В ідеальному випадку, продуктивність дискової підсистеми може бути збільшена в ту кількість разів, яка рівна кількості дисків в масиві. Розмір (ширина) смуги вибирається виходячи з особливостей кожного рівня RAID, і може бути рівна біту, байту, розміру фізичного сектора диска, або ж розміру доріжки. Частіше всього послідовні смуги розділяються по послідовних дисках масиву. Так, в масиві із n дисків n перших смуг фізично розміщені як перші смуги на кожному з дисків, наступні n смуг – як другі смуги на кожному диску і т. д. Набір послідовних смуг, однаково розміщених на кожному диску масиву, називають поясом.

Інша ціль концепції RAID – це підвищення їх відмовостійкості за рахунок виявлення і корекції помилок, які виникають при відмові дисків, або в результаті збой. Досягається це за рахунок надлишкового дискового простору, який задіяний для зберігання додаткової інформації, що дозволяє відновити спотворені або втрачені дані. В RAID передбаченні три види додаткової інформації: дублювання, код Хемінга та використання бітів контролю парності.

Дублювання полягає в записі тих же даних на інші диски масиву, що дозволяє при відмові одного з дисків скористатися відповідною інформацією, збереженою на справних дисках. У принципі розділення інформації по дисках масиву може бути довільним, але для скорочення затрат, зв'язаних із пошуком копії, застосовується розбиття масиву на пари дисків, де в кожній парі дисків інформація є ідентичною та розміщена однаково. При такому дублюванні надмірність дискового масиву становить 100%.

Код Хемінга обчислюється дляожної групи смуг, однаково розміщених на всіх дисках масиву, тобто поясу. Коригуючі біти зберігаються на спеціально виділених для цієї цілі додаткових дисках. Так, для масиву з десяти дисків потрібно чотири додаткових диски.

Використання бітів контролю парності замість коду Хемінга передбачає обчислення контрольної смуги бітів парності для кожного набору смуг, розміщених в ідентичній позиції на всіх дисках масиву. В ній значення окремого біта формується як сума за модулем два для одноименних бітів в усіх контролюваних смугах. Для зберігання смуг бітів контролю парності потрібно тільки один додатковий диск. У випадку відмови якогось із

дисків масиву проводиться звернення до диску бітів контролю парності і дані відновлюються по бітах бітів парності та даних від інших дисків масиву.

Розглянемо шість базових типів дискових масивів RAID: RAID 0, RAID 1, ..., RAID 5 та їх комбінації детальніше.

9.7.2.1. Базовий тип дискових масивів RAID 0

Даний тип RAID побудовано за принципом розшарування пам'яті й орієнтовано на підвищення продуктивності без використання надлишковості (рис. 9.35).

Оскільки RAID 0 не володіє надлишковістю, аварія одного диска приводить до аварії всього масиву. З іншого боку RAID 0 забезпечує максимальну продуктивність й ефективність використання об'єму дисків.

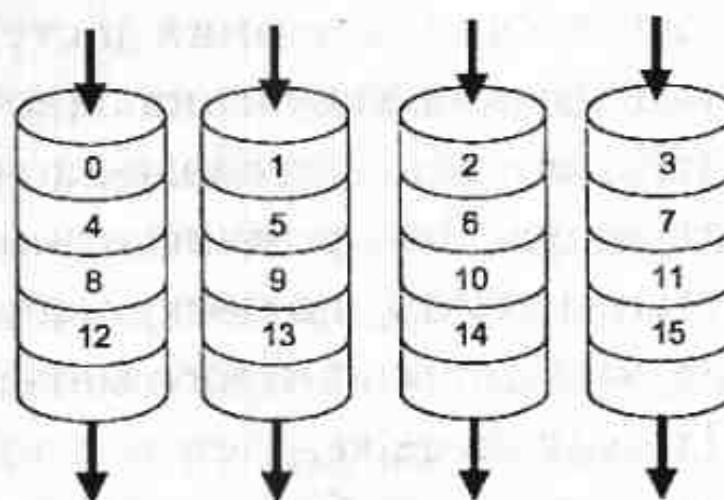


Рис. 9.35. Структура RAID 0

Область застосування: аудіо та відео додатки, що вимагають високої швидкості безперервної передачі даних, яку не може забезпечити один диск. Наприклад, дослідження, проведені фірмою Mylex, з метою визначити оптимальну конфігурацію дискової системи для станції нелінійного відео монтажу показують, що, в порівнянні з одним диском, масив RAID 0 з двох дисків дає приріст швидкості запису-читування на 96%, з трьох дисків – на 143% (за даними тесту Miro VIDEO EXPERT Benchmark). Мінімальна кількість дисків в масиві RAID 0 – 2шт.

9.7.2.2. Базовий тип дискових масивів RAID 1

В даному типі RAID надлишковість досягається дублюванням даних. Тут кожна пара дисків містить однакову інформацію і сприймається як один логічний диск (рис. 9.36).

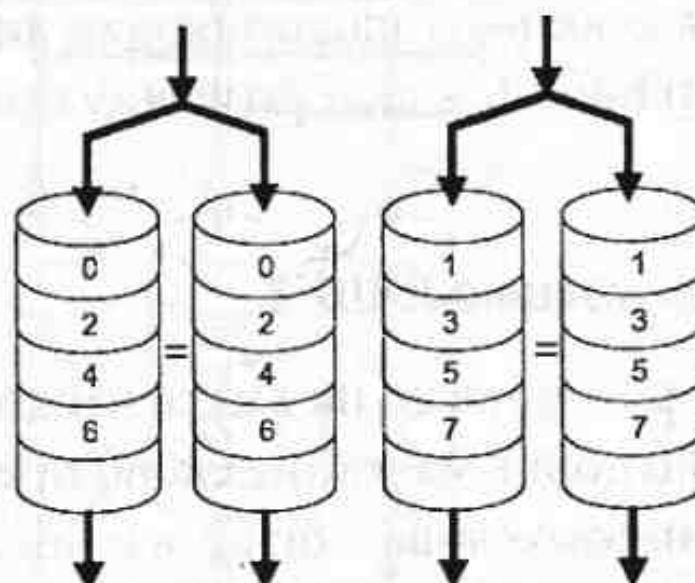


Рис. 9.36. Структура RAID 1

Запис тих же даних проводиться на обидва диски в кожній парі одночасно. Проте, диски, що входять у пару, можуть здійснювати одночасні операції зчитування різних даних. Таким чином, дублювання може подвоювати швидкість зчитування, але швидкість запису залишається незмінною в порівнянні з одним диском. RAID 1 володіє 100% надлишковістю і аварія одного диска не приводить до аварії всього масиву – контролер дисків перемикає операції зчитування/запису на працездатний диск.

RAID 1 забезпечує найвищу продуктивність серед всіх типів надлишкових дискових масивів (RAID 1 – RAID 5), але характеризується найгіршим використанням дискового простору. Мінімальна кількість дисків в масиві RAID 1 – 2шт.

9.7.2.3. Базовий тип дискових масивів RAID 2

У системах RAID 2 використовується паралельний доступ до даних на різних дисках, де у виконанні кожного запиту на введення/виведення одночасно беруть участь всі диски. Шпінделі всіх дисків синхронізовані так, що головки кожного диску в кожен момент часу знаходяться в одинакових позиціях. Дані розбивають на смуги завдовжки в 1 біт і розподіляють по дисках масиву таким чином, що повне машинне слово представляється поясом, тобто число дисків рівне довжині машинного слова в бітах. Для кожного слова обчислюється коригуючий код (зазвичай, це код Хемінга, здатний коригувати одиночні і виявляти подвійні помилки), який, також побітово, зберігається на додаткових дисках (мал. 5.41). Наприклад, для масиву, орієнтованого на зберігання 32-розрядних слів (32 основних диски) потрібно сім додаткових дисків, так як коригуючий код складається з 7 розрядів.

При записі обчислюється коригуючий код, який заноситься на відведені для нього диски. При кожному читанні проводиться доступ до всіх дисків масиву, включаючи додаткові. Зчитані дані разом із коригуючим кодом подаються на контролер дискового масиву, де відбувається повторне обчислення коригуючого коду і його порівняння з тим кодом, що зберігався на надмірних дисках. Якщо присутня одиночна помилка, контролер здатний швидко її розпізнати і виправити, так що час зчитування не збільшується.

RAID 2 дозволяє досягти високої швидкості введення-виведення при роботі з великими послідовними записами, але стає неефективним при обслуговуванні записів невеликої довжини. Основна перевага RAID 2 полягає у високому ступені захисту інформації, проте використаний в цій схемі метод корекції вже існує в диску, оскільки більшість дисків зберігають коди корекції помилок для кожного сектора.

Тому, оскільки використання декількох надлишкових дисків є неефективним, RAID 2 не дає особливих переваг у порівнянні з далі розглянутим RAID 3 і практично не застосовується.

9.7.2.4. Базовий тип дискових масивів RAID 3

Як і у випадку з RAID 2, дані розбивають на смуги завдовжки в 1 біт і розподіляють по дисках масиву таким чином, що повне машинне слово представляється поясом, тобто число дисків рівне довжині машинного слова в бітах, а один з дисків масиву відводиться для зберігання інформації про парність (рис. 9.37). Тобто різниця між RAID 3 і RAID 2 – потрібен лише один додатковий диск для зберігання біту парності, а не декілька дисків для зберігання коду Хемінга.



Рис.9.37. Структура RAID 3

У разі відмови одного з дисків, відновлення інформації, що зберігалася на ньому, можливе шляхом виконання операції виключного АБО (XOR) за інформацією на працездатних дисках. Кожен запис, зазвичай, розподілений по всіх дисках, і тому цей тип масиву дисків є ефективним для роботи в додатках з інтенсивним обміном із дисковою підсистемою. Оскільки кожна операція введення-виведення звертається до всіх дисків масиву, RAID 3 не може одночасно виконувати декілька операцій. Тому RAID 3 хороший для однозадачного оточення з довгими записами. Для роботи з короткими записами потрібна синхронізація обертання дисків, оскільки інакше зменшується швидкість обміну. Застосовується рідко, оскільки програє RAID 5 по використанню дискового простору. Мінімальна кількість дисків в масиві RAID 3 – 3шт.

9.7.2.5. Базовий тип дискових масивів RAID 4

RAID 4 ідентичний RAID 3 за винятком того, що тут розмір поясів є набагато більшим одного сектора. В цьому випадку зчитування можна здійснювати з одного диска (не рахуючи диска, що зберігає інформацію про парність), тому можливе одночасне виконання декількох операцій зчитування (рис. 9.38). Проте, оскільки кожна операція запису повинна відновити вміст диска парності, одночасне виконання декількох операцій записи неможливе. Цей тип масиву не має помітних переваг перед масивом типу RAID 5.

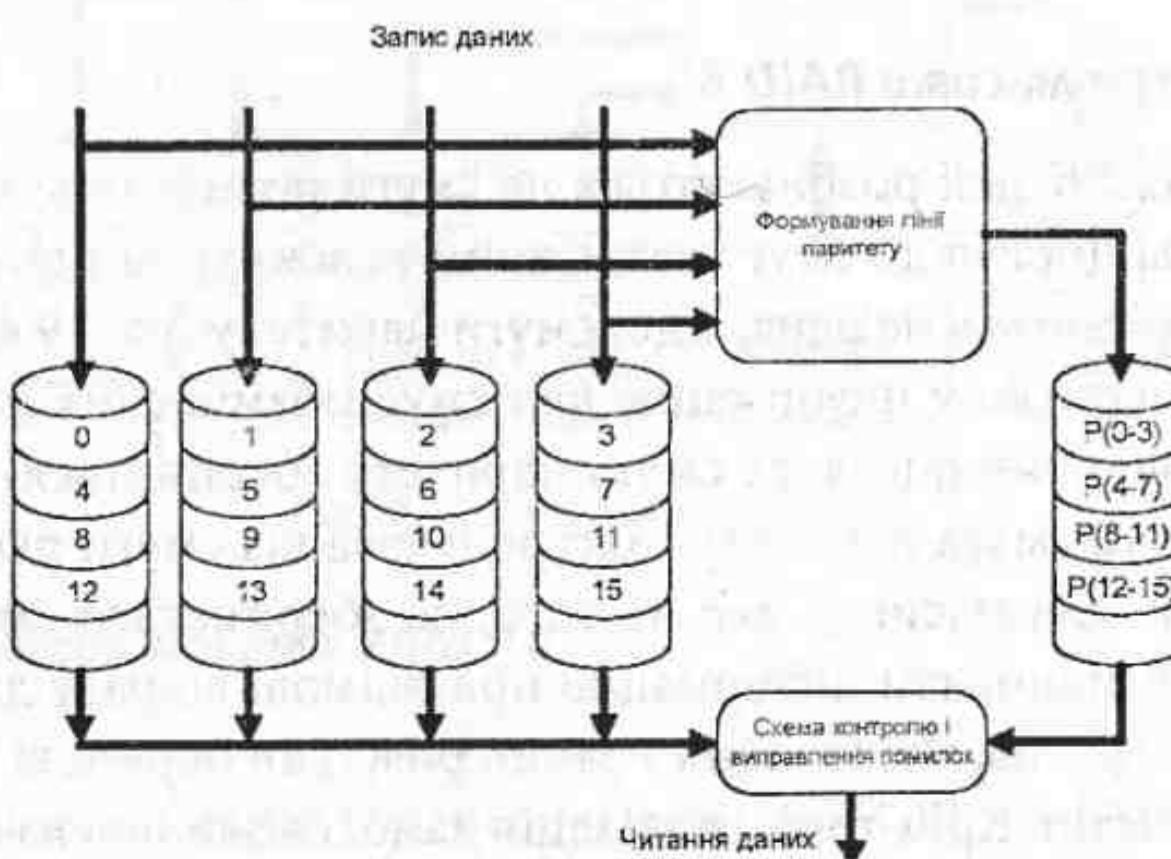


Рис. 9.38. Структура RAID 4

9.7.2.6. Базовий тип дискових масивів RAID 5

Цей тип масиву дисків подібний до типу RAID 4, однак тут немає окремого диску для зберігання бітів парності. Тому тут немає недоліку, властивого RAID 4, який заключається у неможливості одночасного виконання декількох операцій запису. У цьому масиві, як і в RAID 4, використовуються пояси великого розміру, але, на відміну від RAID 4, інформація про парність зберігається не на одному диску, а на всіх дисках по черзі (рис. 9.39).

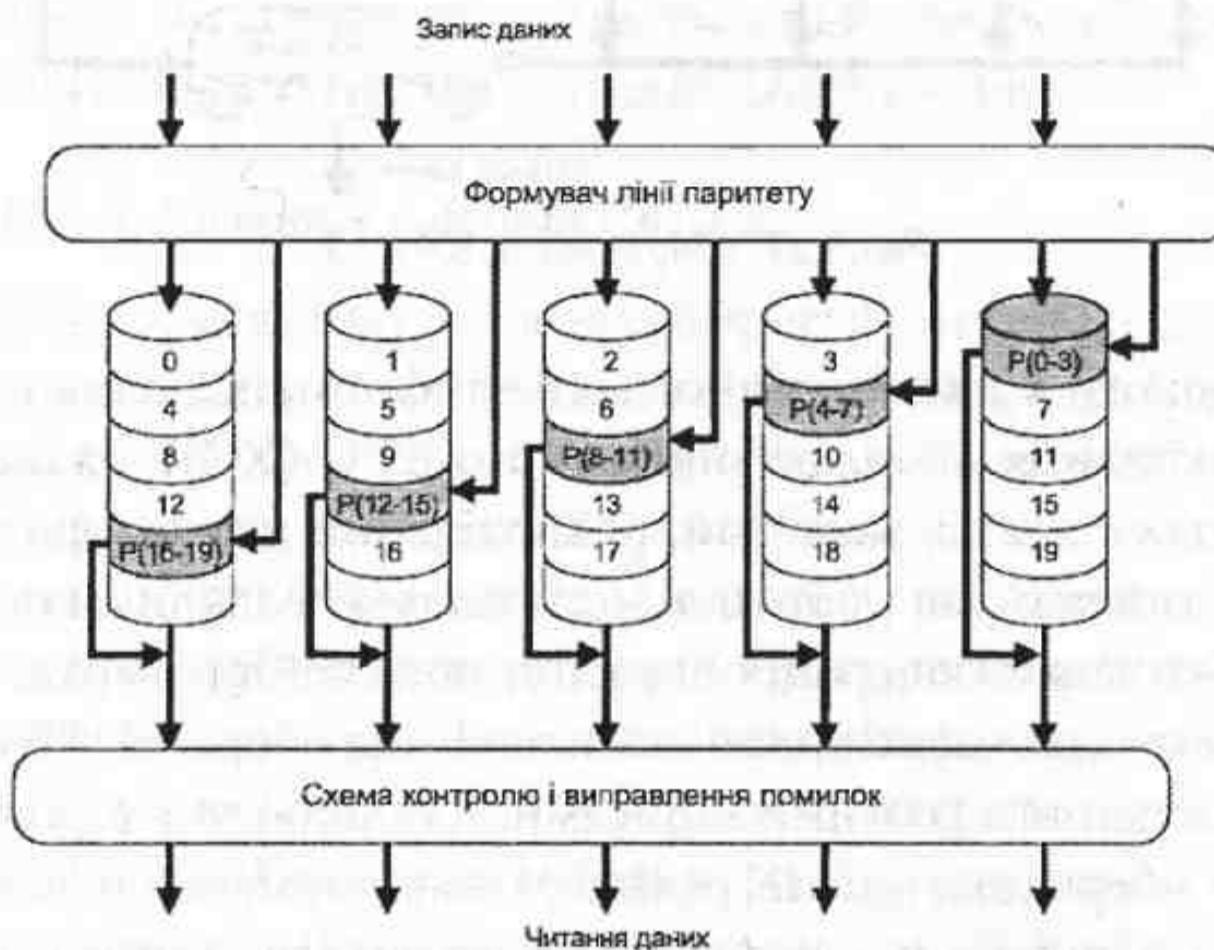


Рис. 9.39. Структура RAID 5

Операції запису звертаються до одного диска з даними і до іншого диска з інформацією про парність. Оскільки біти парності для різних поясів зберігаються на різних дисках, виконання декількох одночасних операцій запису є неможливим, тільки в тих окремих випадках, коли або пояси з даними, або пояси з бітами парності знаходяться на тому ж диску. Чим більше дисків у масиві, тим рідше співпадає місце положення поясу і біта парності.

Область застосування цього типу дискових масивів – надійне зберігання масивів даних великого об'єму. Мінімальна кількість дисків в масиві RAID 5 – 3шт.

9.7.2.7. Тип дискових масивів RAID 6

Як і в RAID 5 в RAID 6 дані розбиваються на смуги розміром в один блок і розподіляються по всіх дисках. Доступ до смуг незалежний та асинхронний. Різниця в тому, що на кожному диску зберігається не одна, а дві смуги паритету (рис. 9.40). Перша з них, як і в RAID 5, містить контрольну інформацію для смуг, розміщених на горизонтальному зразі масиву (за виключенням диска, де смуга паритету зберігається). В додаток формується і записується друга смуга паритету, контролююча всі смуги якогось одного диску масиву. (вертикальний зразок масиву), але не того, де зберігається смуга паритету. Така схема масиву дозволяє відновити інформацію при відмові відразу двох дисків. З іншої сторони, збільшується час на обчислення і запис розрядів парності та вимагається додатковий дисковий простір. Крім того, реалізація даної схеми пов'язана з ускладненням контролера дискового масиву. Тому, даний тип RAID зустрічається рідко.

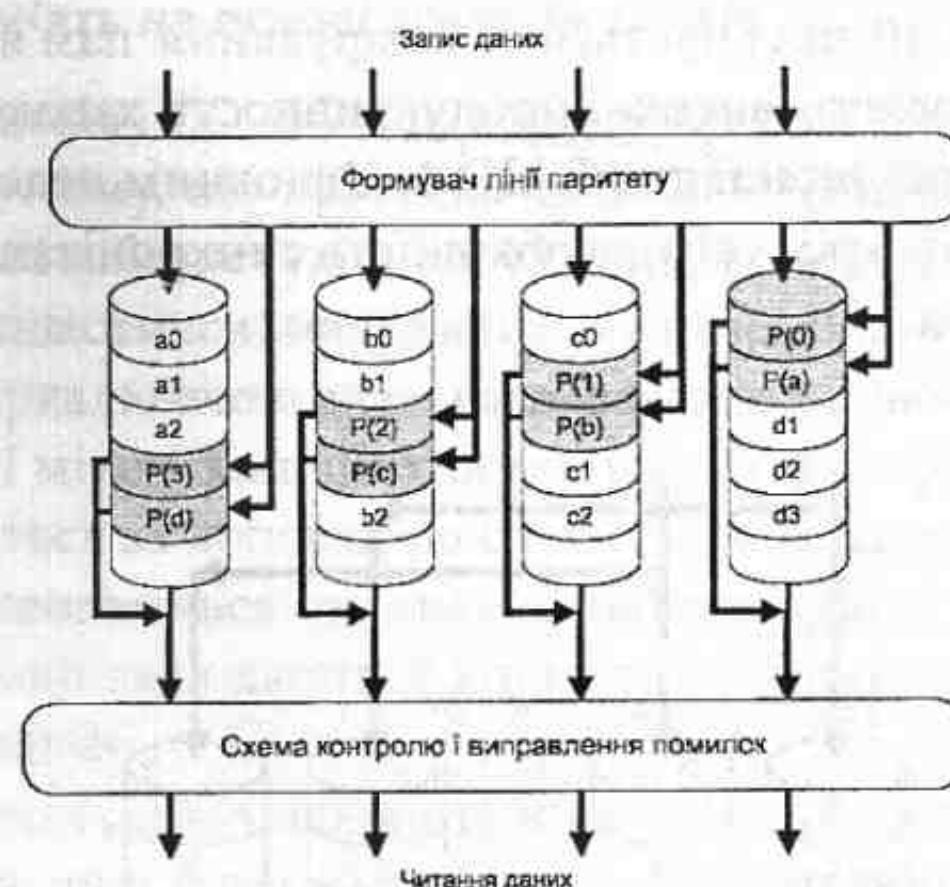


Рис. 9.40. Структура RAID 6

9.7.2.8. Тип дискових масивів RAID 7

RAID 7 не входить до складу основних типів дискових масивів. До складу масиву дисків RAID 7, крім самого масиву асинхронно працюючих дисків, входять кешпам'ять та контролер, які керуються вбудованою в контролер операційною системою реального часу (рис. 9.41). Дані розбиті на смуги розміром у блок і розділені по дисках масиву. Смуги бітів парності зберігаються на спеціально виділених для даної цілі одному або кількох дисках.

Даний тип RAID при роботі з великими файлами не поступається за продуктивністю RAID 3. Разом з тим, RAID 7 може так само ефективно, як і RAID 5, проводити кілька одночасних операцій зчитування і запису для невеликих об'ємів даних, що забезпечується використанням кеш пам'яті і названої операційної системи.

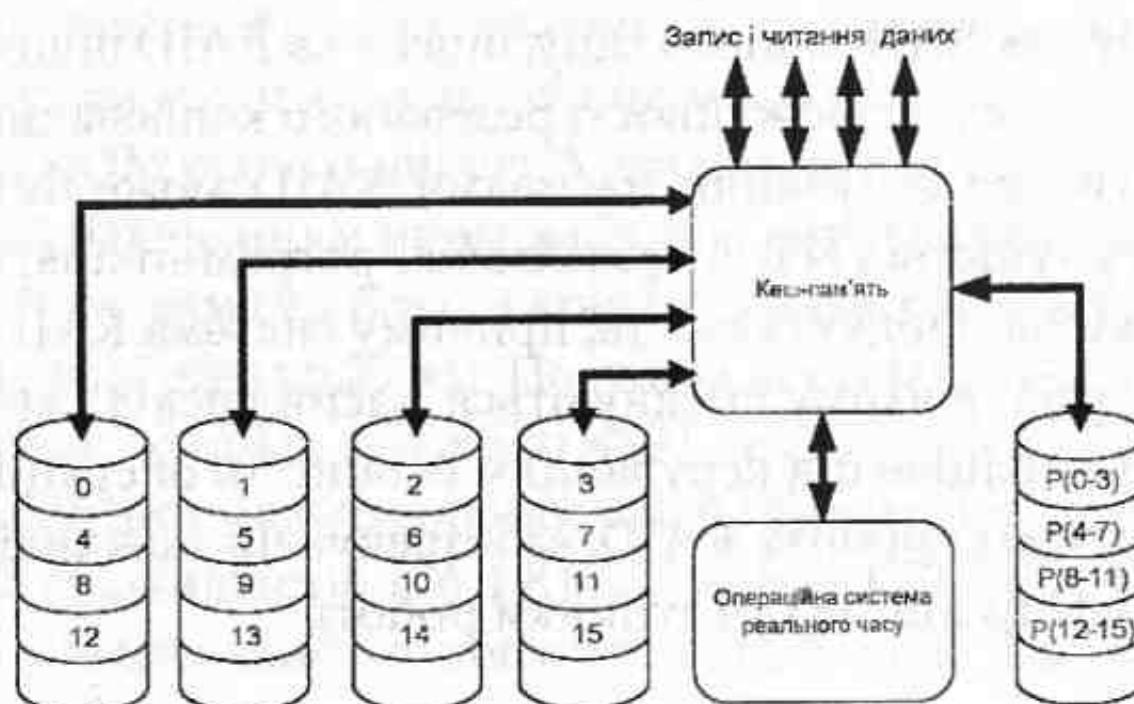


Рис. 9.41. Структура RAID 7

9.7.2.9. Тип дискових масивів RAID 10

RAID 10 також не входить до складу основних типів дискових масивів дана схема співпадає з RAID 0, але, на відміну від неї, роль окремих дисків виконують дискові масиви, побудовані за схемою RAID 1.

Таким чином, в RAID 10 поєднується розшарування пам'яті й дублювання даних (рис. 9.42). Це дозволяє досягти високої продуктивності, характерної для RAID 0, при рівні відмовостійкості, характерної для RAID 1. Основним недоліком цього типу RAID є висока вартість реалізації. Крім того, необхідність синхронізації всіх дисків приводить до ускладнення контролера.

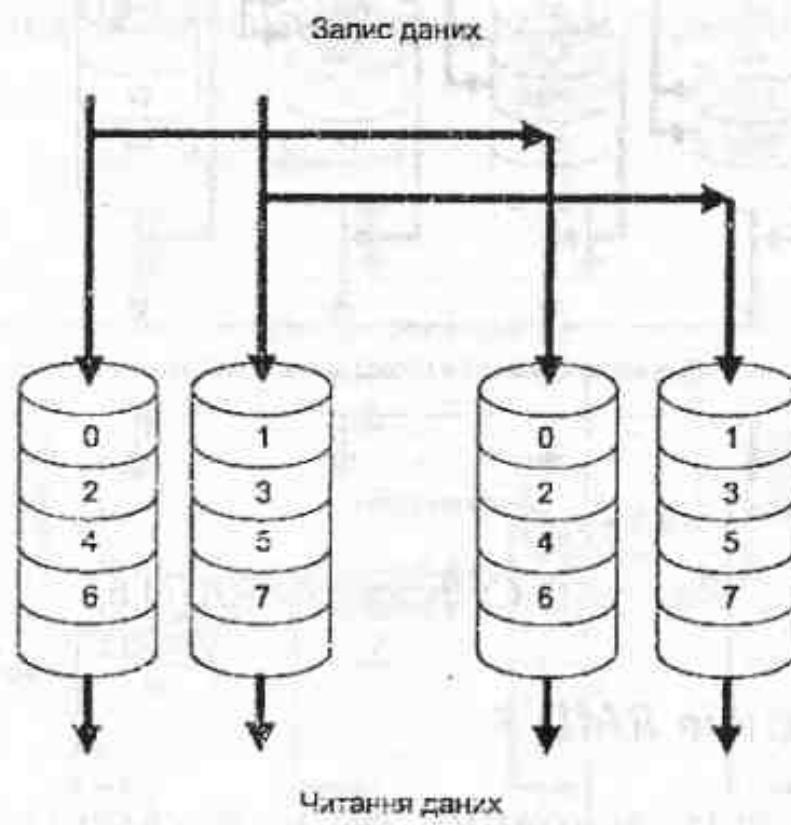


Рис. 9.42. Структура RAID 10

Таку конфігурацію дискових масивів називають дворівневим RAID або RAID 10 (RAID 0+1). Область її застосування: дешеві масиви, в яких головне – надійність зберігання даних.

Як вже було сказано вище, можливі й інші комбінації базових типів дискових масивів. Наприклад, при поєднанні технологій RAID 0 та RAID3, коли в додаток до принципів організації RAID 0 роль окремих дисків виконують дискові масиви, організовані за схемою RAID 3, отримується схема RAID 53. Часто застосовують розділення загального дискового простору на частини, причому об'єм кожної з них визначається користувачем. Одна частина функціонує як RAID одного типу, інша – як RAID іншого типу. Це дозволяє досягти вищої продуктивності та можливості резервного копіювання на випадок збою.

Потрібно відзначити, що керування масивами RAID може бути реалізоване програмно, як, наприклад, у Windows NT, апаратно або програмно-апаратно. При апаратній реалізації досягається вища продуктивність, причому система RAID може бути виконана як автономний пристрій, в якому поєднуються масив дисків і контролер. Контролер містить мікропроцесор і працює під керуванням особистої операційної системи, реалізовуючи різноманітні режими роботи RAID, забезпечуючи можливість заміни несправних дисків без втрати інформації та без зупинки роботи.

9.7.3. Оптична пам'ять

У 1983 році була представлена перша цифрова аудіосистема на базі компакт дисків (CD – compact disk). Компакт-диск – це односторонній диск, здатний зберігати більш ніж 60-хвилинну аудіоінформацію. Величезний комерційний успіх CD сприяв розвитку технології дешевих оптичних запам'ятовуючих пристройів. За подальші роки були створені різні системи пам'яті на оптичних дисках, три з яких у прогресуючому ступені приживаються в комп'ютерах: CD-ROM, WARM і оптичні диски із стиранням.

9.7.3.1. Постійна пам'ять на основі компакт дисків

Для аудіо компакт-дисків і CD-ROM використовується ідентична технологія. Основна відмінність полягає у тому, що програвачі CD-ROM міцніші і містять пристрій для виправлення помилок, які забезпечують коректність передачі даних з диска у комп'ютер. Диск виготовляється з пластмаси, наприклад, полікарбонату, і покритий забарвленим шаром з високою відзеркалюючою здатністю, зазвичай алюмінієм. Цифрова інформація наноситься у вигляді мікроскопічних поглиблень у відзеркалюальній поверхні. Запис інформації проводиться за допомогою сильно сфокусованого променя лазера високої інтенсивності. Так створюється так званий майстер-диск, з якого потім друкуються копії. Поглиблення на копії захищаються від пилу і пошкоджень шляхом покриття поверхні диска прозорим лаком.

Інформація з диска зчитується малопотужним лазером, розташованим у програвачі. Лазер освітлює поверхню диска, що обертається, крізь прозоре покриття. Інтенсивність відбитого променя лазера міняється, коли він потрапляє в поглиблення на диску. Ці зміни фіксуються фотодетектором і перетворюються на цифровий сигнал.

Поглиблення, розташовані ближче до центру диска, переміщаються щодо променя лазера повільніше, ніж більш віддалені. Через це необхідні заходи для компенсації відмінностей в швидкості так, щоб лазер міг прочитувати інформацію з постійною швидкістю.

Одне з можливих вирішень аналогічне до вживаного в магнітних дисках – збільшення відстані між бітами інформації, залежно від її розташування на диску. В цьому випадку диск може обертатися з незмінною швидкістю і, відповідно, такі дискові пристрії відомі як пристрії з постійною кутовою швидкістю. Зважаючи на нераціональне використання зовнішньої частини диска метод постійної кутової швидкості в CD-ROM не підтримується. Натомість інформація по диску розміщується в секторах однакового розміру, які скануються з постійною швидкістю за рахунок того, що диск обертається із змінною швидкістю. В результаті поглиблення прочитуються лазером із постійною лінійною швидкістю. При доступі до інформації у зовнішнього краю диска швидкість обертання є меншою і зростає при наближенні до осі. Ємність доріжки і затримки обертання зростають зі зсувом від центру до зовнішнього краю диска.

Випускаються CD-ROM різної ємності. У типовому варіанті відстань між доріжками складає 1,6 мкм, що, з урахуванням проміжків між доріжками, дозволяє забезпечити 20344 доріжки. Фактично ж, замість безлічі концентричних доріжок, є одна доріжка у вигляді спіралі, довжина якої рівна 5,27 км. Постійна лінійна швидкість CD-ROM – 1,2 м/с, тобто для "проходження" спіралі потрібно 4391 секунд або 73,2 хвилини. Саме ця величина складає стандартний максимальний час програвання аудіодиска. Оскільки дані прочитуються з диска зі швидкістю 176,4 КБ/с, ємність CD-ROM рівна 774,57 МБ.

Дані на CD-ROM організовані як послідовність блоків. Типовий формат блоку показаний на рис. 9.43.



Рис. 9.43. Формат блоку CD-ROM

Блок включає наступні поля:

- Синхронізацію. Це поле ідентифікує початок блоку і складається з нульового байта, десяти байтів, що містять тільки одиничні розряди, і знову байта зі всіх нулів.
- Ідентифікатор. Заголовок, що містить адресу блоку і байт режиму. Режим 0 визначає порожнє поле даних; режим 1 відповідає за використання коду, що коригує помилки, і наявність 2048 байт даних; режим 2 визначає наявність 2336 байт даних і відсутність коригуючого коду.
- Дані. Дані користувача.
- Коригуючий код (КК). Поле призначене для зберігання додаткових даних користувача в режимі 2, а в режимі 1 містить 288 байт коду з виправленням помилок.

Рис. 9.44 ілюструє організацію інформації на CD-ROM. Як вже наголошувалося, дані розташовані послідовно по спіралевидній доріжці. Для варіанту з постійною лінійною швидкістю довільний доступ до інформації стає складнішим.

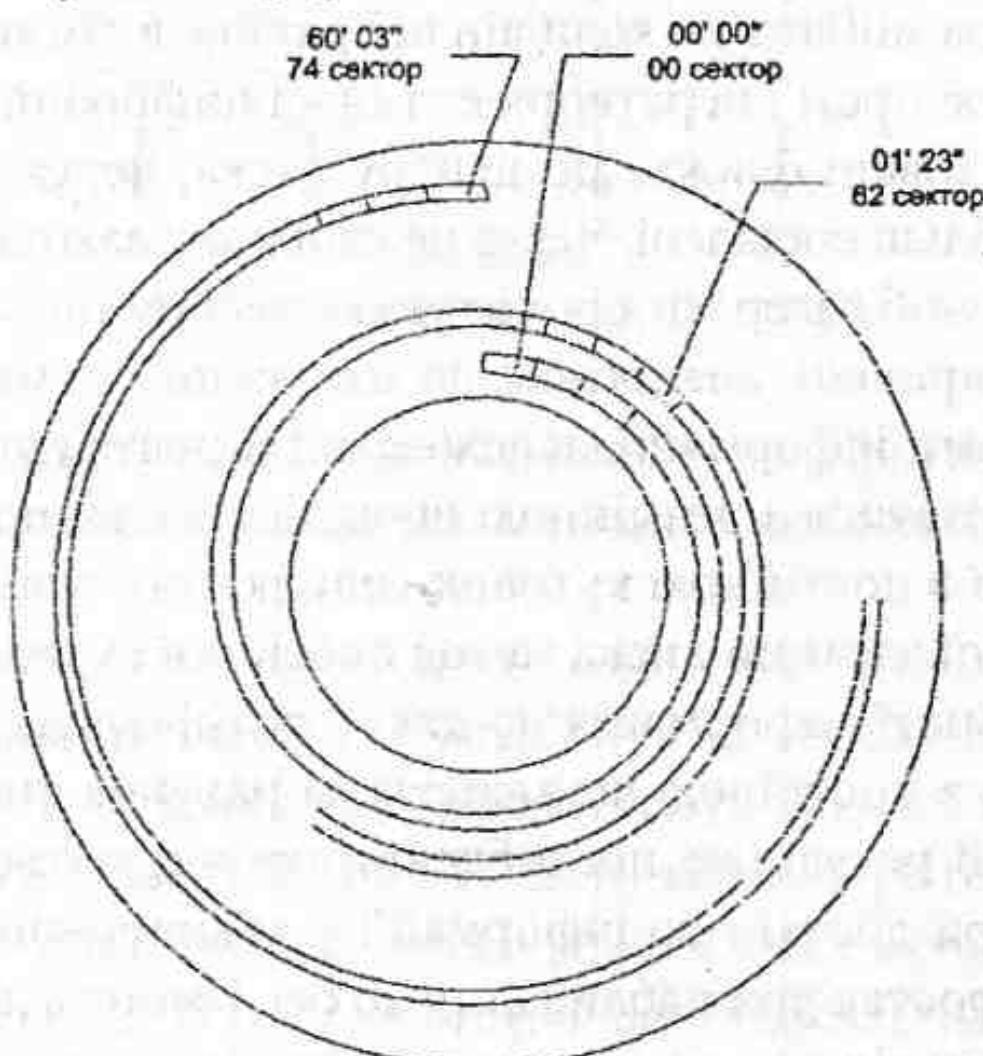


Рис. 9.44. Організація диска з постійною лінійною швидкістю

Останнім часом намітився перехід до нового типу оптичних дисків DVD (Digital Video Data). Диски DVD складаються з двох шарів завтовшки 0,6 мм, тобто мають дві робочих поверхні, і забезпечують зберігання по 4,7 Гбайт на кожній. У технології DVD використовується лазер з меншою довжиною хвилі (650 нм проти 780 нм для стандартних CD-ROM), а також витонченіша схема корекції. Все це дозволяє збільшити число доріжок і підвищити щільність запису. Крім того, при записі застосовується метод компресії інформації, відомий як MPEG2.

9.7.3.2. Оптичні диски із стиранням

Серед багатьох технологій оптичних дисків з можливістю багатократного перезапису інформації комерційно прийнятною виявилася тільки магнітооптична. У таких системах енергія лазерного променя використовується спільно з магнітним полем. Запис і стирання інформації відбуваються за рахунок реверсування магнітних полюсів маленьких областей диска, покритого магнітним матеріалом. Лазерний промінь нагріває

опромінювану пляму на поверхні і у цей момент магнітне поле може змінити орієнтацію магнітних полюсів на опромінюваній ділянці. Оскільки процес поляризації не викликає фізичних змін на диску, йому не страшні багатократні повторення. При читанні напрям магнітного поля можна визначити за поляризацією лазерного променя. Поляризоване світло, відбите від певної плями, змінює свій кут відзеркалення залежно від характеру намагніченості.

9.7.4. Магнітні стрічки

Пам'ять на базі магнітних стрічок використовується в основному для архівації інформації. Носієм служить тонка стрічка полістиролу завширшки 0,38–2,54 см і завтовшки близько 0,025 мм, покрита магнітним шаром. Стрічка намотується на бобіни різного діаметру. Дані записуються послідовно, байт за байтом, від початку стрічки до її кінця. Час доступу до інформації на магнітній стрічці є значно більшим, чим у раніше розглянутих видів зовнішньої пам'яті.

Зазвичай уздовж стрічки розташовується 9 доріжок, що дозволяє записувати упоперек стрічки байт даних і біт парності. Інформація на стрічці групується в блоки – записи. Кожен запис відділяється від сусідньої міжблоковим проміжком, що дає можливість позиціювання головки зчитування/запису на початок будь-якого блоку. Ідентифікація запису проводиться по полю заголовка, що міститься в кожному записі. Для вказівки початку і кінця стрічки використовуються фізичні маркери у вигляді металізованих смужок, що наклеюються на магнітну стрічку, або прозорих ділянок на самій стрічці. Відомі також варіанти маркування початку і кінця стрічки шляхом запису на неї спеціальних кодів – індикаторів.

В комп'ютерах зазвичай застосовуються котушкові пристрої з вакуумними системами стабілізації швидкості переміщення стрічки. У них швидкість переміщення стрічки складає близько 300 см/с, щільність запису – 4 КБ/см, а швидкість передачі інформації – 320 КБ/с. Типова котушка містить 730 м магнітної стрічки.

У пам'яті на базі картриджів використовуються касети з двома котушками, аналогічні стандартним аудіокасетам. Типова ширина стрічки – 8 мм. Найбільш поширеною формою таких пристройів є DAT (Digital Audio Tape). Дані на стрічку заносяться по діагоналі, як це прийнято у відеокасетах. За розміром такий картридж приблизно удвічі менший, ніж звичайна компакт-диск-касeta, і має товщину 3,81 мм. Кожен картридж дозволяє зберігати декілька ГБ даних. Час доступу до даних невеликий (середнє між часами доступу до дискет і до жорстких дисків). Швидкість передачі інформації вища, ніж у дискет, але нижча, ніж у жорстких дисків.

Другим видом пам'яті на базі картриджів є пристрій стандарту DDS (Digital Data Storage). Цей стандарт був розроблений в 1989 році для задоволення вимог до резервного копіювання інформації з жорстких дисків у могутніх серверах і розрахованих на багато користувачів системах. По суті, це варіант DAT, що забезпечує зберігання 2 ГБ даних при довжині стрічки 90 м. У пізнішому варіанті стандарту DDS-DC (Digital Data Storage — Data Compression) за рахунок застосування методів стиснення інформації ємність стрічки збільшена до 8 ГБ. Нарешті, третій тип запам'ятовуючого пристрою на базі картриджів також призначений для резервного копіювання вмісту жорстких дисків, але при менших об'ємах такої інформації. Цей тип відповідає стандарту QIC (Quarter Inch

Cartridge tape) і відоміший під назвою стример. Відомі стримери, що забезпечують зберігання від 15 до 525 МБ інформації. Залежно від інформаційної місткості і фірми-виготовлювача змінюються і характеристики таких картриджів. Так, число доріжок може варіюватися в діапазоні від 4 до 28, довжина стрічки – від 36 до 300 м і т.д.

9.8. Короткий зміст розділу

У даному розділі розглянуті питання організації роботи з пам'яттю комп'ютера. В комп'ютері використовуються різні типи пам'яті, які, залежно від способу доступу до інформації, можуть бути класифіковані наступним чином: пам'ять з довільним доступом, із впорядкованим доступом та з асоціативним доступом. Кожний тип пам'яті має свої переваги та недоліки, які визначають місце його використання в комп'ютері.

Розглянута структура пам'яті комп'ютера та типи пам'яті, які входять до складу внутрішньої та зовнішньої пам'яті комп'ютера. Приведено основні характеристики пам'яті: ємність пам'яті, організація пам'яті, швидкодія пам'яті, час доступу до пам'яті, період звернення до пам'яті, вартість. Названо типи пам'яті залежно від технології виготовлення, енергозалежності, за методом доступу до даних.

Проаналізовано можливі варіанти організації регістрових файлів процесорів. Показано обмеження, які спричиняє використання багатопортового регістрового файла. Розглянуто ряд нових структур регістрового файла: інтегрованого регістрового файла, розподіленого регістрового файла, кластерного розподіленого регістрового файла, розподіленого регістрового файла з керованою комутацією, розподіленого регістрового файла з віконною організацією, ієрархічного регістрового файла. Розглянута динамічна та статична організація даних в регістрових файлах. Описана робота регістрового файла на базі черги з програмованою затримкою.

Описано галузі ефективного використання пам'яті з асоціативним доступом та принципи її роботи, зокрема, як реалізується запис нової інформації та як реалізується зчитування інформації з пам'яті з асоціативним доступом. Показано які види пошуку можна здійснювати в пам'яті з асоціативним доступом. Описано чотири основні елементи організації пам'яті з асоціативним доступом: з повним паралельним асоціативним доступом, з неповним паралельним асоціативним доступом, з послідовним асоціативним доступом, з частково асоціативним доступом.

Описано види запам'ятовуючих пристрій, які може містити основна пам'ять. Приведено можливі варіанти побудови блокової пам'яті та можливості по скороченню часу доступу до інформації, які надає блокова організація пам'яті. Пояснено чим обумовлена ефективність розшарування пам'яті. Показано як здійснюється нарощування розрядності основної пам'яті. Описано топологію запам'ятовуючих елементів, яка лежить в основі організації напівпровідникових ЗП. Дано пояснення призначення керуючих сигналів в мікросхемі пам'яті. Описано як побудовано ПЗП, що запрограмований при виготовленні, одноразово запрограмований після виготовлення та багаторазово програмний ПЗП.

Приведено порядок розміщення інформації на магнітному диску. Описані типи сучасних дискових систем. Пояснено для чого використовуються масиви магнітних дисків з надлишковістю та робота шести базових типів дискових масивів RAID: RAID 0,

RAID 1, ..., RAID 5 та дискових масивів, створених на їх основі. Описана робота оптичної пам'яті та пам'яті на магнітних стрічках.

9.9. Література для подальшого читання

Структура пам'яті комп'ютера та типи пам'яті, які входять до складу внутрішньої та зовнішньої пам'яті комп'ютера приведені в книгах [1–3, 5, 7–11]. Тут же приведено основні характеристики пам'яті: ємність, організація, швидкодія, час доступу, період звернення, вартість. Класифікація різних типів пам'яті залежно від способу доступу до даних приведена в [5]. В роботі [6] було запропоновано новий тип пам'яті, яка за способом доступу до даних відноситься до пам'яті з впорядкованим доступом. Принципи побудови та організації роботи пам'яті з довільним доступом детально розглянуті в літературі [8–11].

В праці [26] проведена класифікація та детально розглянуті структури та принципи організації використовуваних у комп'ютерах регістрових файлів. У роботі [22] описано віконну організацію регістрового файла. Структуру ієрархічного регістрового файла запропоновано в роботі [14]. У роботах [14, 15] пропонується дворівнева модель регістрового файла для динамічних суперскалярних архітектур, яка дозволяє зменшити кількість регістрів та кількість портів. Така дворівнева організація зменшує кількість портів у три рази, а також покращує час доступу до даних на 46%. Регістровий файл процесора SPARC є також ієрархічний, оскільки містить безпосередньо регістрові вікна та асоціативний регістровий кеш [16, 23]. В роботі [25] пропонується будувати регістровий файл на основі черг за принципом FIFO.

В роботах [25, 26] пропонується організувати регістровий файл на основі черги з програмованою затримкою.

Асоціативна пам'ять описана в [1]. Існує цілий ряд алгоритмів, що дозволяють організувати впорядковану вибірку даних з цієї пам'яті. Детальний їх опис і порівняльний аналіз можна знайти в [11].

У 1987 році Паттерсон (Patterson), Гібсон (Gibson) і Катц (Katz) з каліфорнійського університету Берклі опублікували статтю [30]. У цій статті описувалися різні типи дискових масивів, що позначаються скороченням RAID.

9.10. Література до розділу 9

- Искусственный интеллект: В 3-х книгах. Кн. 3. Программные и аппаратные средства: Справочник / Под ред. В. Н. Захарова, В. Ф. Хорошевского. – М.: Радио и связь, 1990. – 191 с
- Каган Б. М. Электронные вычислительные машины и системы. – М.: Энергия, 1979. – 528 с
- Каган Б. М., Каневский М. М. Цифровые вычислительные машины и системы. – М.: Энергия, 1974. – 680 с
- Мельник А. О. Программировані процесори обробки сигналів. – Львів: Видавництво Національного університету “Львівська політехніка”, 2000. – 55 с
- Угрюмов Е. П. Цифровая схемотехника. – СПб.: БХВ – Санкт-Петербург, 2000. – 528 с
- Мельник А. О. Принципи побудови буферної сортувальної пам'яті. Вісник Державного університету “Львівська політехніка” “Комп'ютерна інженерія та інформаційні технології”, N 307, 1996. – С. 65–71.

7. Б Н Малиновский и др. Справочник по цифровой вычислительной технике. – К Техніка, 1980. – 320 с
8. Шигин А Г., Дерюгин А А. Цифровые вычислительные машины (память ЦВМ). – М.: Энергия, 1975. – 536 с
9. D. Patterson, J. Hennessy. Computer Architecture. A Quantitative Approach. Morgan Kaufmann Publishers, Inc. 1996.
10. Patterson, D. A., & Hennessy, J. L. *Computer Organization and Design, The Hardware/Software Interface*, 2nd ed , San Mateo, CA: Morgan Kaufmann, 1997.
11. Метлицкий Б А., Каверзnev В В. Системы параллельной памяти. Теория, проектирование, применение. Под ред. В И. Тихонова. – Л., 1989.
12. David J. Kuck. The Structure of Computers and Computations. John Wiley & Sons, Pittsburgh, Pennsylvania, 1978.
13. S.Y. Kung, VLSI Array Processors, Prentice Hall, 1988.
14. Marcio Merino Fernandes, Josep Llosa, Nigel Topham. Using Queues for Register File Organization in VLIW Architectures. Technical Report ECS-CSG 29-97, Dept of Computer Science, University of Edinburgh, 1997.
15. Henk Corporaal, Microprocessor Architectures: From VLIW to TTA, John Wiley & Sons, Inc., New York, NY, 1997.
16. Greg Blank and Steve Krueger. SuperSPARC: A fully integrated superscalar processor. In Hot Chips III. A Symposium on High-Performance Chips, IEEE, August 1991.
17. CEVA: CEVA-X1620 Datasheet. CEVA, 2005.
18. Texas Instruments: TMS320C64x Technical Overview. 2005. – www.ti.com
19. S. Rixner, W. Dally, B. Khailany, P. Mattson, U. Kapasi. Register organization for media processing. International Symposium on High Performance Computer Architecture (HPCA), pp. 375–386, 2000.
20. Balasubramonian, R., Dwarkadas, S., Albonesi, D. Reducing the Complexity of the Register File in Dynamic Superscalar Processor. In Proceedings of the 34th International Symposium on Microarchitecture, December 2001.
21. R. M. Russell. The CRAY-1 computer system. *Communications of the ACM*, 21(1):63–72, Jan. 1978.
22. R. Ravindran, R. Senger, E. Marsman, G. Dasika, M. Guthaus, S. Mahlke, and R. Brown. Increasing the Number of Effective Registers in a Low-Power Processor Using a Windowed Register File. Proc. 2003
23. David L. Weaver and Tom Germond. The SPARC Architecture Manual, Version9. Sparc International and PTR Prentice Hall, Englewood Cliffs, NJ, 1994.
24. Мельник А О. Спеціалізовані системи реального часу: конспект лекцій. – Львів: навч видання, 1996. – 53 с.
25. Мельник А О., Сало А М. Методика проектування паралельного процесора на основі пам'яті з детермінованою вибіркою // Вісник НУ “Львівська політехніка”. – № 546, 2005. – С. 96–101.
26. Мельник А О., Сало А М. Регістровий файл // Вісник НУ “Львівська політехніка”, 2007. – С. 96–101.
27. Rolf Hakenes. A novel low-power microprocessor architecture. www.iccd-conference.org/proceedings/2000/08010141.pdf
28. Gregory W. A Comparison of Circuits for On-Chip Programmable Crossbar Switches // 10th NASA Symposium on VLSI Design, Albuquerque, NM, March 20–21, 2002.
29. www.xilinx.com
30. Patterson, Gibson, Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID).

9.11. Питання до розділу 9

1. Поясніть принципи організації пам'яті з довільною вибіркою
2. Як зв'язані адреса і ємність пам'яті?
3. Назвіть операції пам'яті
4. Опишіть структуру пам'яті комп'ютера
5. Які типи пам'яті входять до складу внутрішньої пам'яті комп'ютера?
6. Які типи пам'яті входять до складу зовнішньої пам'яті комп'ютера?
7. Які типи пам'яті є в процесорі?
8. Які операції визначає поняття "звернення до пам'яті"?
9. Назвіть основні характеристики пам'яті
10. Які одиниці вимірювання використовуються для вказівки ємності пам'яті?
11. Що таке "організація пам'яті"?
12. Якими характеристиками описується швидкодія пам'яті?
13. Що таке час доступу до пам'яті?
14. Що таке період звернення до пам'яті?
15. У чому полягає відмінність між часом доступу і періодом звернення до пам'яті?
16. Назвіть типи пам'яті залежно від технології виготовлення
17. Які одиниці використовуються для оцінки вартості пам'яті?
18. Назвіть типи енергонезалежної пам'яті
19. Назвіть типи енергозалежної пам'яті
20. Приведіть класифікацію пам'яті за методом доступу до даних
21. Що таке реєстровий файл процесора?
22. Чи є реєстри реєстрового файла програмно доступними? Як це розуміти?
23. Наведіть типи реєстрових файлів
24. Наведіть структуру інтегрованого реєстрового файла
25. Як здійснюється запис даних до інтегрованого реєстрового файла?
26. Як здійснюється зчитування даних з інтегрованого реєстрового файла?
27. На що впливає збільшення кількості портів інтегрованого реєстрового файла?
28. Що таке розподілений реєстровий файл?
29. Приведіть організацію кластерного розподіленого реєстрового файла
30. Яка перевага кластерного розподіленого реєстрового файла в порівнянні з інтегрованим реєстровим файлом?
31. Як організовано кластерний розподілений реєстровий файл процесора TMS320C6xx?
32. Приведіть організацію розподіленого реєстрового файла з керованою комутацією
33. Як працює розподілений реєстровий файл з віконною організацією?
34. Що дає застосування ієрархічного реєстрового файла?
35. Поясніть різницю між динамічною та статичною організацією даних в реєстрових файлах
36. Які переваги в динамічній організації збереження даних в реєстрових файлах порівняно із статичною?
37. Поясніть роботу реєстрового файла на базі черги з програмованою затримкою
38. Де ефективно використовувати пам'ять з асоціативним доступом?
39. Поясніть принципи роботи пам'яті з асоціативним доступом
40. Для чого використовується реєстр збігів у пам'яті з асоціативним доступом?
41. Поясніть призначення маски в пам'яті з асоціативним доступом
42. Як реалізується запис нової інформації в пам'ять з асоціативним доступом?
43. Як реалізується зчитування інформації з пам'яті з асоціативним доступом?
44. Які види пошуку можна здійснювати в асоціативному ЗП?
45. Назвіть чотири основні елементи організації пам'яті з асоціативним доступом

46. Чим відрізняються простий і складний пошуки інформації в пам'яті з асоціативним доступом?
47. Поясніть організацію роботи пам'яті з повним паралельним асоціативним доступом
48. Поясніть організацію роботи пам'яті з неповним паралельним асоціативним доступом
49. Поясніть організацію роботи пам'яті з послідовним асоціативним доступом
50. Поясніть організацію роботи пам'яті з частково асоціативним доступом
51. Які види запам'ятовуючих пристройів може містити основна пам'ять?
52. Охарактеризуйте можливі варіанти побудови блокової пам'яті
53. Які можливості по скороченню часу доступу до інформації надає блокова організація пам'яті?
54. Чим обумовлена ефективність розшарування пам'яті?
55. Як здійснюється нарощування розрядності основної пам'яті?
56. Яка топологія запам'ятовуючих елементів лежить в основі організації напівпровідникової ЗП?
57. Яку мінімальну кількість ліній повинен містити стовпець МС пам'яті?
58. Поясніть призначення керуючих сигналів в мікросхемі пам'яті
59. Чим обумовлена необхідність регенерації вмісту динамічних ОЗП?
60. Охарактеризуйте основні сфери застосування статичних і динамічних ОЗП
61. Який вид ПЗП має найвищу швидкість перепрограмування?
62. Як побудовано ПЗП, що запрограмований при виготовленні?
63. Як побудовано ПЗП, який одноразово запрограмований після виготовлення?
64. Приведіть структуру матриці одноразово програмованого ПЗП
65. Як побудовано багаторазово програмований ПЗП?
66. Назвіть типи багаторазово програмованих ПЗП
67. Як розміщена інформація на магнітному диску?
68. Назвіть типи сучасних дискових систем
69. Для чого використовуються масиви магнітних дисків з надлишковістю?
70. Поясніть роботу шести базових типів дискових масивів RAID: RAID 0, RAID 1, ..., RAID 5 та дискових масивів, створених на їх основі
71. Назвіть типи оптичної пам'яті
72. Як організована пам'ять на магнітних стрічках?