

БУФЕРНА ПАМ'ЯТЬ КОМУТАТОРІВ

В статті проаналізовані методи комутації, схеми буферизації та розподілу буферної пам'яті, типи пам'яті, їх характеристики та сформовані вимоги до пам'яті комутатора, вибрана як буферна багатопортова сортувальна пам'ять.

Ключові слова: *комутатор, пам'ять, архітектура, комутація, доступ, буфер.*

Вступ. Інфраструктура широкосмугових комунікацій базується на взаємозв'язаних мережах різного масштабу [1,2]. Незалежно від протоколів верхнього рівня - на найнижчому знаходяться високошвидкісні комутатори (switches). Комутатор здійснює комутацію інформаційних потоків, з'єднуючи відповідні вхідні та вихідні порти. Головними вимогами до комунікаційного обладнання є збільшення пропускної спроможності та поліпшення інших його характеристик і мережі в цілому. Сучасна комутація використовує високошвидкісні інтерфейси з продуктивністю внутрішньої комутаційної матриці в декілька сотень Гбіт/с [2, 107]. Необхідність забезпечення можливості статистичного мультиплексування потоків, що проходять через комутаційні системні модулі, передача різних видів трафіку з несхожими вимогами до кількісних характеристик функціонування мережі (доля втрачених даних, допустимий відсоток помилок, час затримки) є непростими завданнями. Комутатори складаються з інтегрованих модулів, які передають потік даних, буферизують, маршрутизують, управляють трафіком, окремими з'єднаннями, мережею.

Одним з важливих вузлів комутатора є пам'ять. Розвиток елементної бази комунікаційного обладнання вніс суттєві зміни в будову пам'яті, що значно покращило її технічні характеристики. При роботі в режимі реального масштабу часу на межі продуктивності потрібно буферизувати дані, що надходять з багатьох каналів, одночасно з комутацією та відправкою раніше прийнятих даних. Для існуючих типів пам'яті ці завдання є складними через їх потенційні обмеження. Тому виникає потреба в створенні та використанні пам'яті з новими властивостями.

Завданнями даного дослідження є аналіз методів комутації, схем буферизації та розподілу буферної пам'яті, типів пам'яті, їх характеристик, формулювання вимог до пам'яті комутатора, вибір принципів проектування буферної пам'яті.

Методи комутації, схеми буферизації, схеми розподілу буферної пам'яті, типи пам'яті взаємозалежать один від одного.

Методи комутації, які визначають структуру комутатора поділяють на наступні [3,119]:

- З розподіленою пам'яттю. Вона повинна мати продуктивність $2NV$ біт/с (N – кількість каналів) при швидкості надходження даних по каналу V біт/с. Обмеження продуктивності пам'яті усувається використанням паралельної організації. Розмір пам'яті залежить від кількості вхідних і вихідних каналів N , інтенсивності надходження вхідних даних і від характеру розподілу вхідних запитів. В загальному випадку пам'ять може бути організована як розподілена чи суміщена.

- З розподіленим середовищем передачі (кільце, шина, подвійна шина). Всі вхідні пакети синхронно комутуються на високопродуктивну паралельну шину, що працює зі швидкістю в $N+1$ раз більшою ніж частота надходження вхідних даних. Вихідний інтерфейс складається з фільтра адрес та пам'яті типу FIFO. Основна проблема - це високопродуктивна шина та буферні елементи зі швидкодією $(N+1)*V$, де V – швидкодія порту. Перевагою є максимальна пропускна здатність при мінімальному часі комутації.

- З повнозв'язною топологією. Переваги в буферизації блоків даних на вихідних портах і відсутності обмежень на групову і широкомовну передачі. Метод допускає масштабування в широких межах і дозволяє досягти високої швидкості функціонування комутатора, оскільки всі його апаратні модулі працюють з однаковою швидкістю. Однак, квадратичне зростання числа буферів обмежує кількість вихідних портів, хоча швидкість обміну через порт лімітується лише фізичною швидкодією адресних фільтрів і вихідних буферів.

- З просторовим поділом (з єдиним і множинними шляхами від вхідного порту до вихідного). Для мережі бан'яного типу критичною є наявність внутрішнього блокування. Рішеннями є розміщення буферів в точках конфлікту чи перед комутуючою структурою розміщують паралельний пристрій сортування за алгоритмом Бетчера.

Ситуація, при якій декілька блоків вхідних даних одночасно мають бути зкомутувані на один вихідний порт називається вихідним конфліктом. Комутаційне поле є внутрішньо блокуючим, якщо набір з N блоків даних, адресованих на N різних вихідних портів, здатний викликати конфлікти при передачі через це поле [4]. Наявність внутрішніх блокувань призводить до зниження максимально можливої продуктивності.

Для кожної з архітектур комутаційного поля необхідна буферизація. Найпоширенішими варіантами розміщення буферів в комутаторі можна вважати 4 наступних:

- організація черг на вході - небезпека виникнення блокування на початку черги. Якщо два блока даних, що одночасно поступили, прямують на один і той же вихідний порт, один з них попаде у вхідний буфер і перешкоджатиме проходженню наступних за нею блоків даних, знижуючи тим самим пропускну спроможність комутатора;

- організація черг на виході - оптимальна за продуктивністю та затримками, але вимагає вживання додаткових засобів для організації одночасної множинної доставки блоків даних на будь-який вихідний порт

- внутрішня організація черги - може стати причиною небажаних випадкових затримок при проходженні блоків даних структури комутатора;

- вживання рециркуляційних буферів – дозволяє блокам повторно проходити по мережі з просторовим розподілом, коли декілька блоків даних одночасно адресуються на один вихідний порт, причому блокувані комірки, прямують на вхідні порти мережі через рециркуляційні буфери. Однак, ємкість комутаційного поля має бути достатньою для розміщення рециркульованих блоків і ускладнюється управління комутацією через необхідність зберегти вихідну послідовність блоків даних.

Кожен з них має сильні і слабкі сторони, проте сьогодні перевага віддається організації черг на виході комутатора. Зазначимо, що з однієї сторони тип комутатора однозначно визначає множину ефективних в поєднанні з ним типів буферної пам'яті, а з іншої – тип буферної пам'яті задає можливі типи комутаторів.

Число і розмір буферів також важливі при розробці комутатора. Необхідно виділити п'ять схем розподілу буферної пам'яті:

- схема повного розподілу (CP), передбачає фіксоване розділення сумарної ємкості буферного простору (може фізично об'єднувати різні буфери) між даними, що направляються на різні виходи;

- повнодоступна схема (CS) – вхідні комірки буферизуються за наявності вільного місця в загальній пам'яті, незалежно від адреси виходу;

- повнодоступна схема з індивідуальними обмеженнями на довжини вихідних черг (SMQ) - число PDU кожного типу не може перевищувати заданого значення (межі), тобто за наявності загальної повнодоступної буферної пам'яті вводяться фіксовані максимальні значення кількості буферизованих блоків даних, що направляються на різні виходи;

- неповнодоступна схема (SMA) є комбінацією CP і CS, бо передбачається загальна буферна пам'ять (CS) і її виділені частини для блоків даних кожного типу (CP);

- неповнодоступна схема з індивідуальними межами (SMQMA) - відрізняється від попередньої введенням індивідуальних обмежень в загальній частині буферної пам'яті.

Аналіз типів пам'яті. У мікропроцесорній техніці використовують різні типи пам'яті, які можна класифікувати за способом доступу до даних наступним чином [5,309]:

- Пам'ять із впорядкованим доступом. Із пам'яті дані вибираються в порядку, який визначається її внутрішньою структурою.

- Пам'ять з довільним доступом. До/з довільної комірки в кожному такті може записуватись/зчитуватись дане, використовуючи адресу комірки на адресному вході пам'яті.

- Пам'ять з асоціативним доступом. Пошук даних здійснюється шляхом порівняння асоціативних ознак, які ставляться у відповідність кожному елементу даних, що зберігається в пам'яті. Пошук завершується видачею елемента даних із потрібною асоціативною ознакою.

Цей список може бути розширений іншими типами пам'яті, які ще не знайшли широкого застосування, наприклад, пам'яттю з програмованим доступом, коли записані до пам'яті дані зчитуються в наперед заданому порядку.

Проаналізуємо ці типи пам'яті для формулювання вимог до пам'яті комутатора.

З пам'яті з впорядкованим доступом дані вибираються в порядку, який визначається її внутрішньою структурою. Пам'ять з впорядкованим доступом будується для зчитування/записування даних через вхідний регістр, а правило їх переміщення між комірками

пам'яті апаратно закріплене в зв'язках між ними. До такої пам'яті належить зокрема пам'ять з послідовним доступом, широко використовувана в мікропроцесорній техніці, з якої дані зчитуються послідовно одне за одним в порядку їх запису або в порядку зчитування.

Пам'ять з послідовним доступом будується так, що дані зчитуються/записуються з/до цієї пам'яті послідовно 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). Побудована на регістрах пам'ять з послідовним доступом часто називається стеком.

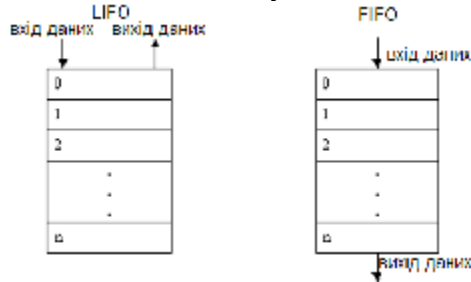


Рис. 1. Принцип організації пам'яті з послідовним доступом

У першому типі пам'яті нове слово заноситься у верхню комірку, раніше занесені дані проштовхуються вниз. При зчитуванні навпаки, останнє слово виштовхується вгору першим.

У разі організації типу FIFO нове слово заноситься у верхню комірку, раніше записані слова виштовхуються вниз. Частіше використовують пам'ять типу "останній прийшов, перший вийшов". Вона організовується наступним чином:

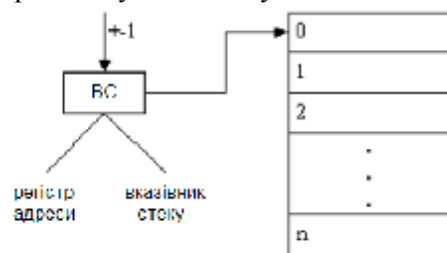


Рис. 2. Адресація стека типу LIFO з допомогою вказівника стеку (BC)

Перед початком роботи в покажчик стеку заноситься початкова адреса. Подальша адресація здійснюється автоматично при виконанні операції запису - читання шляхом збільшення - зменшення адреси на одиницю. Фізичний процес запису/читання даних відбувається так, як і в пам'яті з довільним зверненням.

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

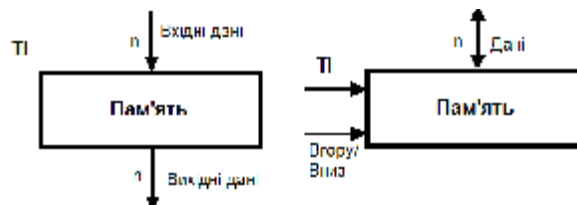


Рис. 3. Інтерфейс пам'яті з послідовним доступом з розділеними та об'єднаними входом і виходом

Перевагою пам'яті є швидкість та простота реалізації. Недоліком 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 довільним доступом (адресна 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. Слова складаються із бітів із значеннями 0 або 1. В слові є n біт, де n - довжина слова. Кожен біт має свій номер. Нумерація біт в слові здійснюється справа-наліво, або навпаки. Зазвичай слово має довжину $n = 2^k$, де $k = 0, 1, 2, \dots$ біт.

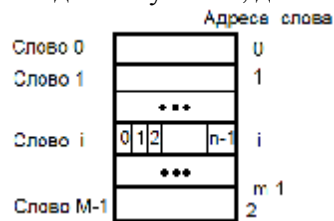


Рис. 4. Адресна пам'ять організацією $n \cdot 2^m$

Комірки пам'яті нумеруються, тобто кожна з них має свій номер (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. 4). M адрес є адресним полем (простором). Використовуючи двійкове кодування, необхідно m біт для представлення всіх адрес, де $m = \lceil \log M \rceil$. 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. Організація пам'яті вказує на розрядність комірок і їх кількість в пам'яті, тобто $n \cdot 2^m$. Це значення вказує також ємність пам'яті в бітах.

Пам'ять з довільним доступом виконує дві основні операції: вибірку Fetch (зчитування Read) і запам'ятовування Store (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 дані, і сигналом зчитування це дане зчитується з вказаної адресою комірки пам'яті (рис. 5).

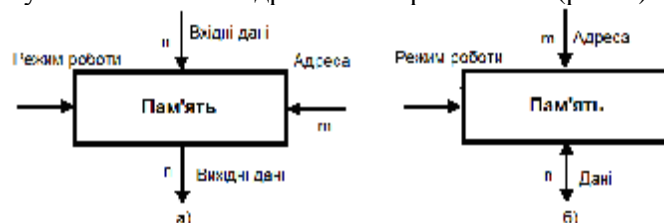


Рис. 5. Інтерфейс адресної пам'яті

Інтерфейс адресної пам'яті (рис. 5 а) включає m -розрядну шину адреси, n -розрядну шину вхідних даних, n -розрядну шину вихідних даних та однорозрядний вхід задання режиму роботи: запису/зчитування. Зазвичай використовуються також вхід дозволу використання пам'яті та вхід для подачі тактових імпульсів, необхідних для синхронізації роботи пам'яті. В інтерфейсі адресної пам'яті (рис. 5 а) шини вводу та виводу даних розділені. Часто пам'ять будується з об'єднаними шинами вводу та виводу даних (шина даних тут є двонаправленою) (рис. 5 б), що спрощує використання такої пам'яті в одношинних і багатошинних структурах.

Перевагою даного типу пам'яті є простота та легкість реалізації. Недоліками є неможливість розпаралелювання роботи (створення паралельної багатопортової пам'яті), створення складної системи адресації та методів взаємодії з нею, великі затрати на елементи доступу до кожної комірки з входу та виходу пам'яті чи з об'єднаного входу/виходу та сповільнення швидкодії.

Пам'ять з асоціативним доступом (асоціативна пам'ять) зберігає разом з даними і їх ознаки, в ролі якої може бути і саме дане. Ядро пам'яті з асоціативним доступом показано на рис. 6. Роль комірок цієї пам'яті виконують регістри. Числа записуються в довільні вільні регістри пам'яті. Дані вибираються з такої пам'яті на основі збігу їх ознак з заданою. Для цього ознаки даних з усіх регістрів пам'яті поступають на схему порівняння, де порівнюються з заданою ознакою із регістра ознаки, і на вихід пам'яті поступають дані, ознаки яких збігаються з заданою.

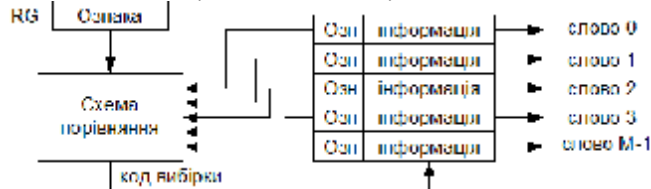


Рис. 6. Ядро пам'яті з асоціативним доступом

В паралельній асоціативній пам'яті одночасно порівнюються з аргументом всі розряди всіх полів ознак пам'яті (одночасний пошук по слову). В послідовній асоціативній пам'яті одночасно порівнюються з бітом аргументу по одному біту кожного поля ознаки (порозрядний пошук).

Перевагою є можливість паралельної роботи (одночасно зчитати всі дані з однаковими ознаками). Недоліками є складність механізму одночасного доступу з кожного каналу до кожної комірки пам'яті з одночасним порівнянням всіх ознак у комірках з шуканими ознаками та забезпечення такого пошуку при великих ємностях пам'яті, сповільнення швидкодії через двократне звертання до пам'яті при операціях запису та читання, великі затрати на елементи доступу до кожної комірки з входу та виходу пам'яті чи з об'єднаного входу/виходу з одночасним пошуком даних, що сповільнює роботу пам'яті.

Принципи побудови буферної пам'яті (БП). Головною функцією БП є забезпечення одночасного прийому вхідних даних з декількох вхідних каналів, їх затримка на потрібну кількість тактів впорядкування за заданим законом та видача декількома вихідними каналами. У більшості випадків БП повинна бути багатоканальною, причому число входів і виходів може бути різним, мати великі ємність та швидкодію.

Коли потрібно затримати дані без їх перестановки, застосовується стекова пам'ять типу FIFO. Вона може бути побудованою на основі регістрів або на основі оперативної пам'яті з довільним доступом. В першому випадку послідовно з'єднуються m (по розміру стека) регістрів. В цій БП з кожним тактовим імпульсом в перший регістр стека приймається нове число, прийняті раніше числа зсуваються на один крок по послідовно з'єднаних регістрах, а з останнього регістра знімається число прийняте найпершим. При побудові стекової пам'яті з програмованою величиною затримки регістри стека з'єднуються між собою комутуючою мережею для виключення частини регістрів, щоб створити стек меншого ніж m розміру, та з'єднати вхідний регістр створеного стека з входом БП, а вихідний регістр створеного стека - з виходом БП.

При побудові стекової пам'яті з величиною затримки m на основі пам'яті з довільним доступом необхідні два оперативні запам'ятовуючі пристрої (ОЗП) ємністю m слів кожний, лічильник і тригер. В кожний момент часу один ОЗП працює в режимі зчитування, а другий - запису. В лічильнику знаходиться значення величини затримки. Його виходи з'єднані з адресними входами обох ОЗП. Тактовими імпульсами вхідні дані записуються в один із ОЗП, в цей час з іншого ОЗП зчитуються вихідні дані, а від лічильника віднімається одиниця. Тим самим змінюється адресація комірок пам'яті. Як тільки вміст лічильника дорівнює нулю, тригер перемикається в інший стан. Сигнал з його виходу змінює режим роботи обох ОЗП і записує в лічильник нове значення m . Порівняно з регістровою, в стековій пам'яті на основі ОЗП необхідно в два рази більше запам'ятовуючих елементів, а її швидкодія менша. Разом з тим, в обох цих типах пам'яті не повністю враховуються наведені вище вимоги до БП, в першу чергу не виконується впорядкування даних у вихідному масиві.

Якщо в розглянутій БП на основі ОЗП, крім затримки масивів даних, необхідно виконувати впорядкування даних, користувач, як правило, задає номери комірок пам'яті запису та зчитування відповідно вхідних та вихідних чисел. Однак робити це немає необхідності, оскільки БП можна організувати так, що користувачу буде достатньо задати лише одну адресу, яка вказує порядковий номер числа, яке приймається у вихідному масиві. Всі необхідні додаткові перетворення повинна виконувати БП, яка названа сортувальною, тому що вона сортує дані у вихідному масиві за значеннями їх адрес. БП реалізується двома основними підходами: на базі ОЗП і на базі процесора сортування чисел (ПСЧ) не рекурсивного типу.

При першому підході можна запропонувати структуру БП (рис. 8). При роботі БП із різними частотами запису та зчитування в схему слід додати комутатор для подачі відповідних тактових імпульсів на входи ОЗП1 і ОЗП2, а також розділити тактові входи регістра R_A і лічильника ЛТ.

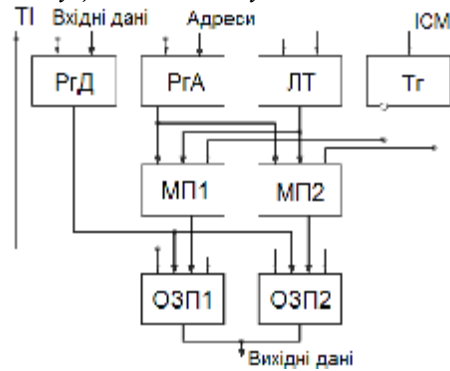


Рис. 8. Буферна сортувальна пам'ять на базі ОЗП

До недоліків БП на базі ОЗП потрібно віднести велику інерцію, оскільки почати зчитування із ОЗП1 або ОЗП2 можна тільки після завершення прийому всього масиву даних, а також сповільнення роботи при наявності багатьох входів і виходів.

Функції БП і функції процесора сортування чисел (ПСЧ) досить подібні: БП виконує сортування елементів масиву даних у відповідності до величини адрес, які їх супроводжують, а ПСЧ сортує числа в порядку їх зростання чи зменшення. Ця подібність нашоухує на думку про можливість використання принципів побудови ПСЧ при синтезі БП. Використовуючи розглянуті базові структури БП, можна запропонувати велику кількість схем їх реалізації. При цьому їх конкретний варіант визначається структурою ПСЧ.

Висновки

В даному дослідженні були проаналізовані методи комутації, наведені схеми буферизації та розподілу буферної пам'яті, досліджені типи пам'яті, встановлені їх характеристики. Сформульовані вимоги до пам'яті комутатора дозволили вибрати принципи проектування буферної пам'яті. При побудові комутаторів доцільно використовувати відповідні алгоритми та пристрої планування комутації при використанні сортувальної пам'яті.

Перспективи. Подальші дослідження будуть направлені на синтез структур запам'ятовуючих пристроїв на основі базових типів буферної сортувальної пам'яті.

1. Н. Jonathan Chao. Broadband Packet Switching Technologies: A Practical Guide to ATM Switches and IP Routers / Н. Jonathan Chao, Cheuk H. Lam, Eiji Oki. John Wiley & Sons Inc, 2001. - 459 с.
2. Cisco Systems, Inc. Руководство по технологиям объединенных сетей, 4-е издание. : Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1040 с.: ил.
3. Каун Ю.В. Постановка задач комутації в мережах / Ю.В. Каун, М.Я Максимович // Наукові нотатки. ЛНТУ. - 2009. - №26. - с. 116-122.
4. Дунець Р. Б. Топології комп'ютерних систем. - Львів: Ліга-Прес, 2007 - 46 с
5. Мельник А.О. Архітектура комп'ютера. – Луцьк: Волинська обласна друкарня, 2008. – 470 с.
6. Мельник А.О. Принципи побудови буферної сортувальної пам'яті // Вісн. Держ. ун-ту "Львівська політехніка" "Комп'ютерна інженерія та інформаційні технології", 1996. – №307. – С.65–71.