

Розділ 12

Паралельні комп'ютерні системи

До цих пір ми знайомилися в основному з однопроцесорними комп'ютерами з архітектурою Джона фон Неймана. Однопроцесорні комп'ютери досягли високих показників продуктивності завдяки технологічним та архітектурним досягненням, які були детально розглянуті в попередніх розділах. Але ріст їх продуктивності стримується як фізичними, технологічними обмеженнями, так і обмеженими можливостями покращення традиційної архітектури Джона фон Неймана. Як ми бачили з попередніх розділів, такі архітектурні рішення, як спрощена система команд, конвеєрна обробка команд та даних, суперконвеєрна та суперскалярна обробка команд, використання довгого формату команди, векторна обробка команд та природний паралелізм, практично виводять однопроцесорні комп'ютери до граничної межі, коли використовуються всі потенційні можливості паралелізму архітектури комп'ютера Джона фон Неймана.

У цьому розділі будуть розглянуті питання подальшого підвищення продуктивності комп'ютерів шляхом створення паралельних комп'ютерних систем. До таких систем належать як багатопроцесорні комп'ютерні системи, в яких використовується паралелізм розподілу задач на велику кількість вузлів обробки, так і комп'ютерні мережі, які мають іншу форму паралелізму – мережу структурно автономних комп'ютерів. Деякі з розглянутих підходів уже знайшли місце в сучасних комп'ютерах. Інші підходи розглядаються як основа для покращення характеристик перспективних комп'ютерів.

12.1. Використання принципів паралельної обробки інформації в архітектурі комп'ютера

Сьогодні використання принципів паралельної обробки інформації в архітектурі комп'ютерів є звичним. Практично всі сучасні комп'ютери використовують той або інший вид паралельної обробки інформації. На презентаціях нових мікросхем і в інформаційних матеріалах виробників це подається як останнє слово техніки і передовий край науки, і це дійсно так, якщо розглядати реалізацію цих принципів у мініатюрних рамках одного кристалу.

Водночас, ідеї паралельної обробки інформації з'явилися досить давно. Вони впроваджувалися в найбільш передових, дорогих, а тому одиничних, комп'ютерах свого часу. Пізніше, після належного відпрацювання технології і здешевлення виробництва, вони використовувалися в комп'ютерах середнього класу, і нарешті сьогодні вони в повному обсязі втілюються у робочих станціях і персональних комп'ютерах.

Прослідкуємо за впровадженням значущих нововведень в архітектурі комп'ютера з точки зору паралельної обробки інформації, починаючи практично з часу створення перших комп'ютерів.

Паралельна пам'ять та паралельна арифметика. Всі найперші комп'ютери (EDSAC, EDVAC, UNIVAC) мали послідовну пам'ять, із якої слова зчитувалися послідовно біт за бітом. Першим комерційно доступним комп'ютером, що використовував паралельну пам'ять і паралельну арифметику, став комп'ютер IBM 701, який поступив на ринок у 1953 році. В більш популярній моделі IBM 704, що поступила на ринок у 1955 році, крім названого, була вперше застосована пам'ять на феритових сердечниках і апаратний арифметичний пристрій з рухомою комою.

Незалежні процесори введення-виведення. В перших комп'ютерах процесори не лише обробляли інформацію, а й керували введенням-виведенням. Проте швидкість роботи найшвидшого зовнішнього пристрою, а в ті часи це була магнітна стрічка, була в 1000 разів меншою швидкості процесора, тому під час операцій введення/виведення процесор фактично простоював. У 1958 р. до комп'ютера IBM 704 приєднали 6 незалежних процесорів введення/виведення, що після одержання команд від центрального процесора могли працювати паралельно з ним, а сам комп'ютер перейменували в IBM 709. Дані модель була досить вдалою, підтвердженням чого є те, що було продано біля 400 її екземплярів, які знаходились в експлуатації близько 20 років.

Передбачення та розшарування пам'яті. У 1961 році фірма IBM закінчила розробку комп'ютера STRETCH, що мав дві принципово важливі особливості: передбачення для вибірки команд і розшарування пам'яті на два блоки для узгодження низької швидкості вибірки з пам'яті та швидкості виконання операцій.

Конвеєр команд. Вперше конвеєрний принцип виконання команд був використаний у комп'ютері ATLAS, розробленому в Манчестерському університеті в 1962 році. Виконання команд розбито на 4 стадії: вибірка команди, обчислення адреси операнда, вибірка операнда і виконання операції. Конвеєризація дозволила зменшити час виконання команд із 6 мкс до 1,6 мкс. Створення даного комп'ютера зробило величезний вплив як на подальший розвиток архітектури комп'ютера, так і на розвиток системного програмного забезпечення: у ньому вперше використана мультипрограмна операційна система, що використовувала віртуальну пам'ять і систему переривань.

Незалежні операційні пристрої. В 1964 році фірма Control Data Corporation (CDC) при особистій участі одного з її фундаторів Сеймура Крея випускає комп'ютер CDC-6600 – перший комп'ютер, у якому використовувалося декілька незалежних операційних пристроїв. Для порівняння із сьогоднішнім днем приведемо деякі характеристики цього комп'ютера:

- час такту рівний 100 нс;
- продуктивність рівна 2–3 млн операцій за секунду;
- основна пам'ять розбита на 32 блоки по 4096 60-розрядних слів;
- цикл пам'яті рівний 1 мкс;
- 10 незалежних операційних пристроїв.

Комп'ютер CDC-6600 мав великий успіх на ринку, суттєво потіснивши комп'ютери фірми IBM.

Конвеєрні незалежні операційні пристрої. В 1969 році фірма CDC випускає комп'ютер CDC-7600 із вісмома незалежними конвеєрними операційними пристроями, тим

самим забезпечивши поєднання паралельної та конвеєрної обробки даних. Основні характеристики цього комп'ютера:

- такт рівний 27,5 нс;
- продуктивність рівна 10–15 млн операцій за секунду;
- 8 конвеєрних операційних пристрій;
- дворівнева пам'ять.

Матричні процесори. В 1967 році були розпочаті роботи над створенням матричного процесора ILLIAC IV, проект якого передбачав досягнення наступних характеристик: 256 процесорних елементів (ПЕ) – 4 квадранти по 64 ПЕ, з можливістю реконфігурації на 2 квадранти по 128 ПЕ, або на 1 квадрант із 256 ПЕ, такт рівний 40 нс, продуктивність рівна 1 млрд. операцій з рухомою комою за секунду. В якості пристрою керування був використаний універсальний комп'ютер із невеликою продуктивністю. Кожний ПЕ мав власний АЛП з повним набором команд та пам'ять ємністю 2К 64-розрядних слів з циклом 350 нс. Комп'ютер був введений в експлуатацію у 1974 році в складі лише 1 квадранта, такт 80 нс, реальна продуктивність наблизилась до 50 млн операцій з рухомою комою за секунду. Однак даний проект суттєво вплинув на архітектуру наступних комп'ютерів, побудованих за схожим принципом, які дістали назву масивно-паралельних комп'ютерів із розподіленою пам'яттю. Ідея побудови комп'ютерів цього класу наступна: беруться серійні мікропроцесори зі своєю локальною пам'яттю та з'єднуються за допомогою деякого комунікаційного середовища. Переваги такої архітектури наступні: якщо потрібно підвищити продуктивність, то збільшується кількість процесорів, якщо обмежені фінанси або заздалегідь відома необхідна продуктивність, то легко підібрати оптимальну конфігурацію і т. п.

Проте є і суттєвий недолік. Справа в тому, що міжпроцесорна взаємодія в комп'ютерах цього класу йде набагато повільніше, ніж відбувається локальне опрацювання даних самими процесорами. Саме тому написати ефективну програму для таких комп'ютерів дуже складно.

За цим принципом були побудовані комп'ютери PEPE, BSP, ICL DAP. До даного класу можна також віднести комп'ютери Intel Paragon, IBM SP1, Parsytec, у якісь мірі IBM SP2 і CRAY T3D/T3E. До цього ж класу можна віднести і мережі комп'ютерів.

Векторно-конвеєрні комп'ютери. У 1972 році С. Крей залишає фірму CDC і засновує власну компанію Cray Research, що у 1976 р. випускає перший векторно-конвеєрний комп'ютер CRAY-1 з наступними характеристиками: час такту рівний 12,5 нс, 12 конвеєрних операційних пристрій, пікова продуктивність – 160 мільйонів операцій за секунду, основна пам'ять до 1М слова (слова 64-розрядні), цикл пам'яті рівний 50 нс. Головним нововведенням є використання векторних команд, що працюють із масивами цілих чисел і дозволяють ефективно використовувати конвеєрні операційні пристрій, а також ієрархія пам'яті. Ієрархія пам'яті прямого відношення до паралелізму не має, проте, безумовно, належать до тих особливостей архітектури комп'ютерів, що мають велике значення для підвищення їхньої продуктивності (згладжування різниці між тактом роботи процесора і часом вибірки з пам'яті). Основні рівні: регістри, кеш пам'ять, основна пам'ять, дискова пам'ять. Час вибірки по рівнях пам'яті від дискової до регістрів зменшується, вартість у перерахунку на 1 слово (байт) росте. В даний час подібна ієрархія підтримується і на персональних комп'ютерах.

Паралельні комп'ютери із спільною пам'яттю. З розвитком архітектур та технологій виробництва компонентів комп'ютера, в тому числі пам'яті, з'явився новий клас комп'ютерів – паралельні комп'ютери із спільною пам'яттю. Вся основна пам'ять таких комп'ютерів розділяється декількома однаковими процесорами.Хоча кількість процесорів, що мають доступ до спільної пам'яті, за чисто технічними причинами не можна зробити великим, до даного напрямку входить багато сучасних багатопроцесорних систем, наприклад, комп'ютери HP Exemplar і Sun StarFire.

Зрозуміло, що в багатьох випадках у комп'ютерах використовується комбінація описаних технічних рішень. З декількох процесорів (традиційних або векторно-конвеєрних) і спільної для них пам'яті формується обчислювальний вузол. Якщо отриманої обчислювальної потужності недостатньо, то об'єднується декілька таких вузлів швидкісними каналами зв'язку. Подібну архітектуру називають кластерною, і за таким принципом побудовані комп'ютери CRAY SV1, HP Exemplar, Sun StarFire, NEC SX-5, останні моделі IBM SP2 й інші.

Системи з масовою паралельною обробкою інформації. Після початкового домінування систем, що використовували повільні процесори з складною системою команд та розшаровану пам'ять, фірми Sun, HP, DEC, SGI, NCR/ATT, Tandem, Pyramid, IBM та CRI розпочали випуск багатопроцесорних систем, здатних до масштабування, що базуються на процесорах із складною системою команд. Наведемо приклади сучасних систем цього класу.

Суперком'ютер Intel Paragon. На рис. 12.1. подано організацію суперком'ютера Intel Paragon із так званою архітектурою на основі пересилання повідомлень. Суперком'ютер побудовано як тривимірне об'єднання множини окремих вузлів (топологія гіперкуба). Кожен вузол містить два або більше процесорів i860, які утворюють систему з спільною пам'яттю, та вбудовану підтримку зовнішньої мережі із швидкодією обміну з мережею 175 MB за секунду. Кожен вузол має ще індивідуальний зв'язок із власною підсистемою пам'яті з використанням прямого доступу та шини кеш пам'яті. Це дозволяє ефективно пересилати велики пакети інформації від кожного вузла до мережі та зворотно. Продук-

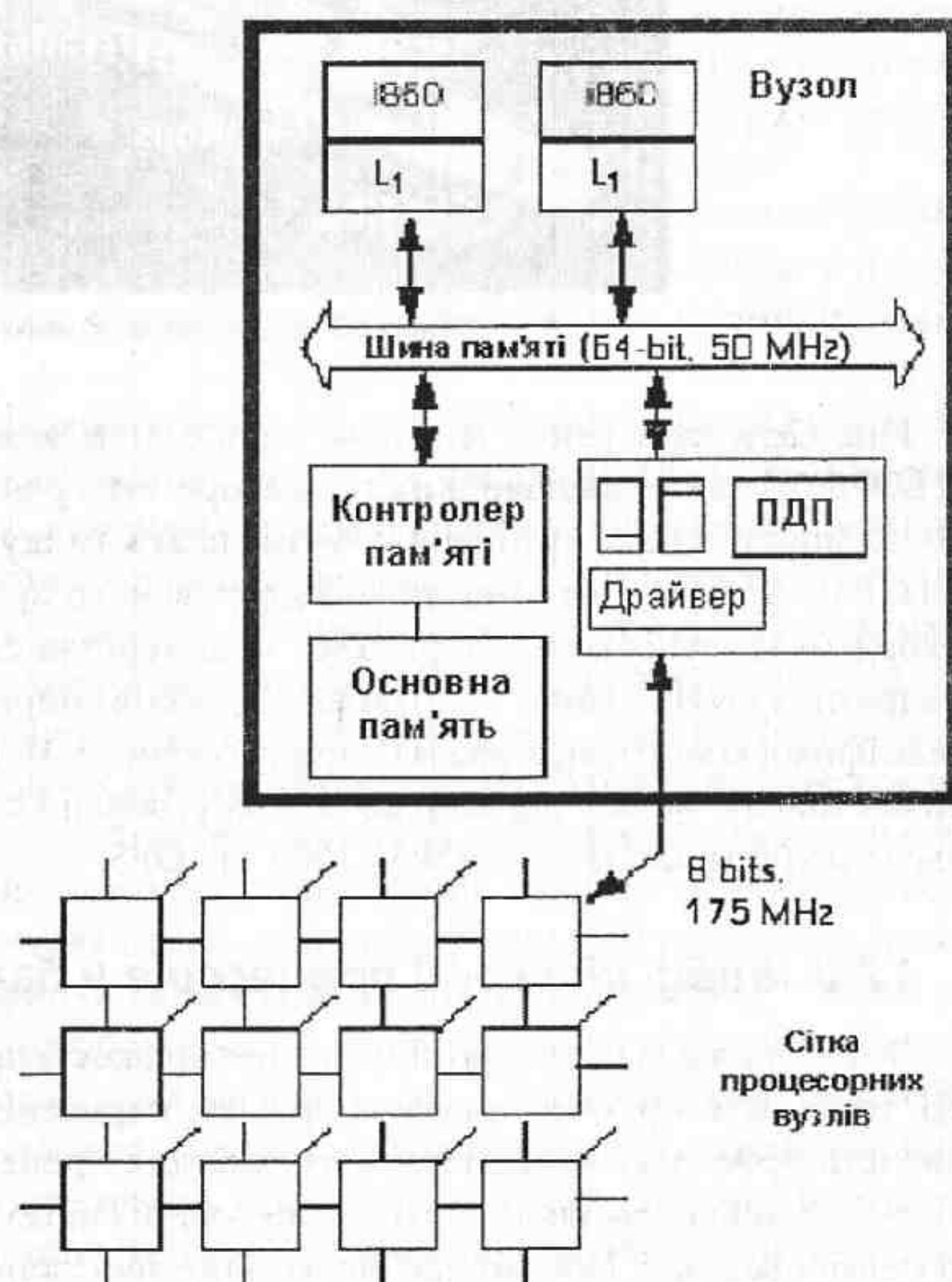


Рис. 12.1. Суперком'ютер Intel Paragon
(подано двовимірну проекцію тривимірної мережі)

тивність системи складає 143 GFLOPS на тесті LINPACK (матричні тести лінійної алгебри), що на порядок перевищує продуктивність машин CRAY.

Суперкомп'ютер Blue Gene фірми IBM. Найшвидший на 2007 рік суперкомп'ютер, створення якого фірма IBM проголосила у грудні 1999 року, отримав назву Blue Gene. Він спроможний виконувати понад один квадрильйон операцій на секунду (1PFLOPS = 1000 TFLOPS = 1000000 GFLOPS). Blue Gene є майже в мільйон разів продуктивнішим від ПК. Ієрархічний дизайн Blue Gene наведено на рис. 12.2.

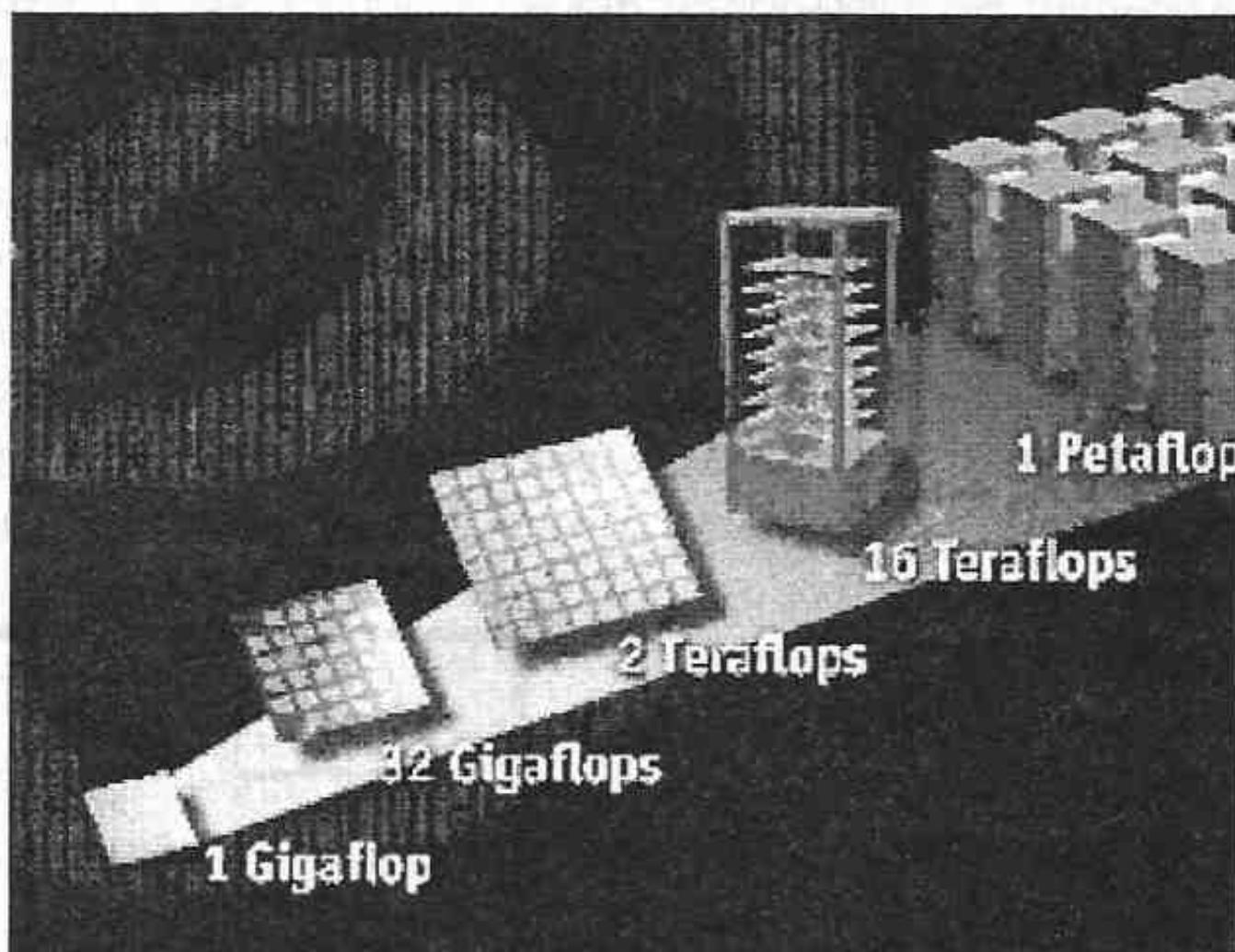


Рис. 12.2. Ієрархічний дизайн Blue Gene

Blue Gene складено з мільйона процесорів, кожний з яких має продуктивність 1 GFLOPS. 32 таких надшвидких процесори інтегровано до одного кристалу (32 GFLOPS). Компактна (два тути на два тути) плата вміщує 64 щойно зазначених кристали (2 TFLOPS). Апаратна частина такого друкованого вузла є продуктивнішою від відомого рекордсмена, суперкомп'ютера ASCI міністерства енергетики США, який займає 8000 квадратних футів площини. 11 друкованих вузлів збирають до одної монтажної 6-ти футової шафової конструкції, яка має продуктивність 16 TFLOPS. Фінальну конструкцію, що займає площину, меншу від 2000 кв. футів, утворено сполученням 64 шаф. Разом отримується продуктивність 1 PFLOPS = 1000 TFLOPS.

12.2. Вибір кількості процесорів у багатопроцесорній системі

Потрібно відзначити, що збільшення кількості процесорів у багатопроцесорній системі не завжди пропорційно зменшує час вирішення задачі. На рис. 12.3 наведено залежність прискорення від кількості задіяних процесорів при розв'язанні однієї задачі молекулярної динаміки на багатопроцесорній системі Intel Paragon. Параметром є версія програмного коду. Бачимо, що монотонне зростання прискорення досягнуто лише на другій версії паралельної програми.

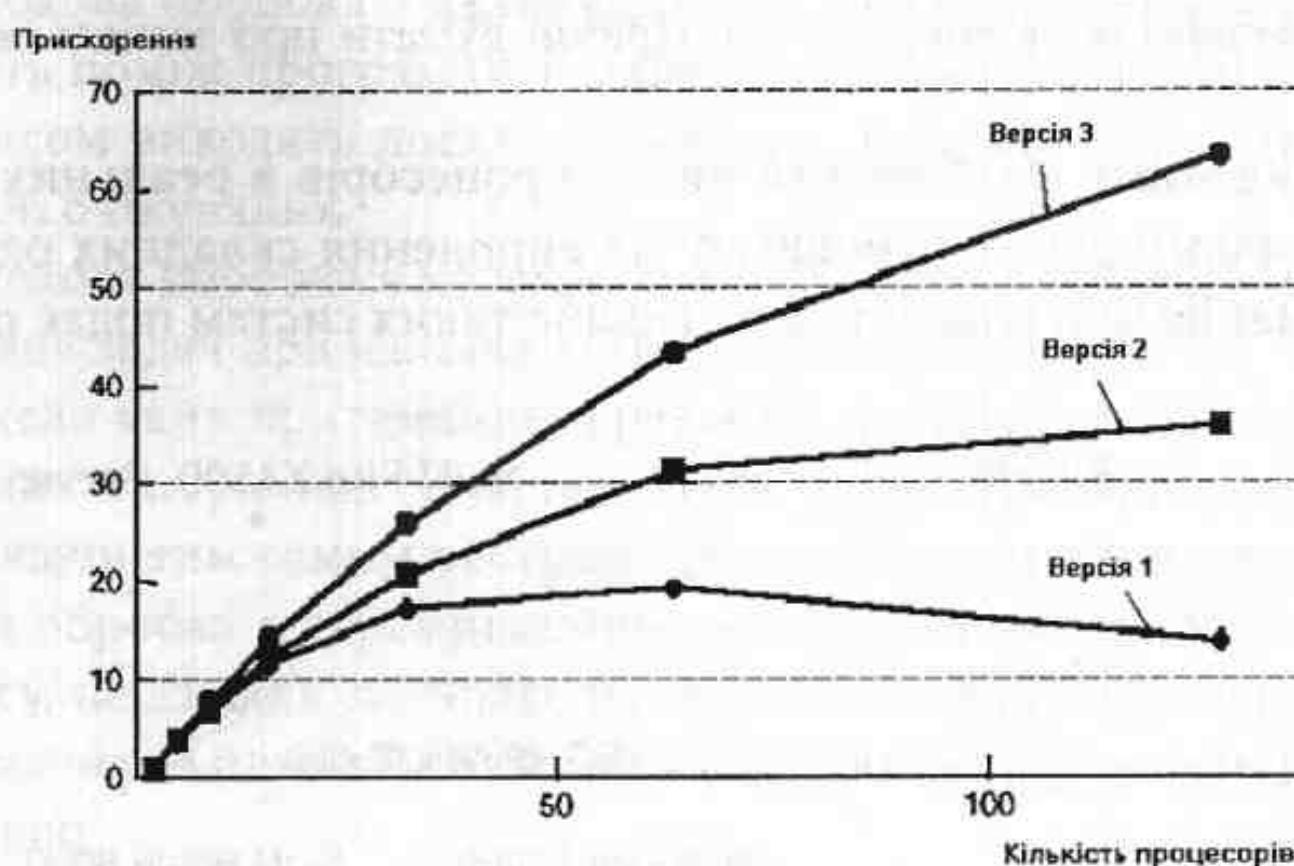


Рис. 12.3. Прискорення програми моделювання процесів молекулярної динаміки для системи Intel Paragon

Перша версія програми погано піддалася розпаралелюванню. З невеликими витратами, шляхом детальнішого налагоджування коду фрагментів програми, що опрацьовувалися окремими процесорами, отримано другу версію. Та лише радикальне покращення програмної частини реалізації інформаційних обмінів окремими процесорами надало остаточний варіант за версією 3, із наближенням до лінійного прискорення.

Зрозумілим є намагання програмістів забезпечити лінійне прискорення (теоретична межа), коли із збільшенням у p разів кількості задіяних процесорів також у p разів зменшується час виконання програми. Однак це не завжди вдається, і причину цього пояснює закон Амдала.

Закон Амдала і його наслідки. Припустимо, що у деякій програмі частка операцій, які потрібно виконувати послідовно, дорівнює f , де $0 \leq f \leq 1$ (при цьому частка розуміється не по статичному числу рядків коду, а по числу реально виконуваних операцій). Крайні випадки в значеннях f відповідають цілком паралельним ($f=0$) і цілком послідовним ($f=1$) програмам. Отож, для того, щоб оцінити, яке прискорення S може бути отримане на комп’ютері з p процесорами при даному значенні f , можна скористатися законом Амдала, згідно з яким $S = p/(1+(p-1)f)$. При великих значеннях p прискорення S прямує до величини $1/f$. Тобто, відповідно до цього закону, якщо $9/10$ програми здійснюється паралельно, а $1/10$, як і раніше, послідовно, то прискорення більш ніж у 10 разів одержати в принципі неможливо поза залежністю від якості реалізації паралельної частини коду і кількості використовуваних процесорів (ясно, що прискорення в 10 разів досягається лише в тому випадку, коли час виконання паралельної частини дорівнює 0).

Подивимося на проблему з іншої сторони: а яку ж частину коду треба прискорити (а значить і попередньо досліджувати), щоб одержати задане прискорення? Відповідь можна знайти в наслідку із закону Амдала: для того, щоб прискорити виконання програми в q разів необхідно прискорити не менше, ніж у q разів $(1-1/q)$ -у частину програми. Отже, якщо є бажання прискорити програму в 100 разів у порівнянні з її послідовним варіантом, то необхідно в 100 разів прискорити 99% коду, що майже завжди складає значну частину програми. Таким чином, якщо після оцінки закладеного в програмі алгоритму стало зрозуміло, що частка послідовних операцій велика, то на значне прискорення

розраховувати безумовно не доводиться і потрібно думати про заміну окремих компонент алгоритму.

У зв'язку з наведеними проблемами число процесорів в реальних багатопроцесорних системах, які орієнтовані в основному на вирішення складних одиничних задач, є невеликим. Інформацію про кількість процесорів таких систем подає рис. 12.4.

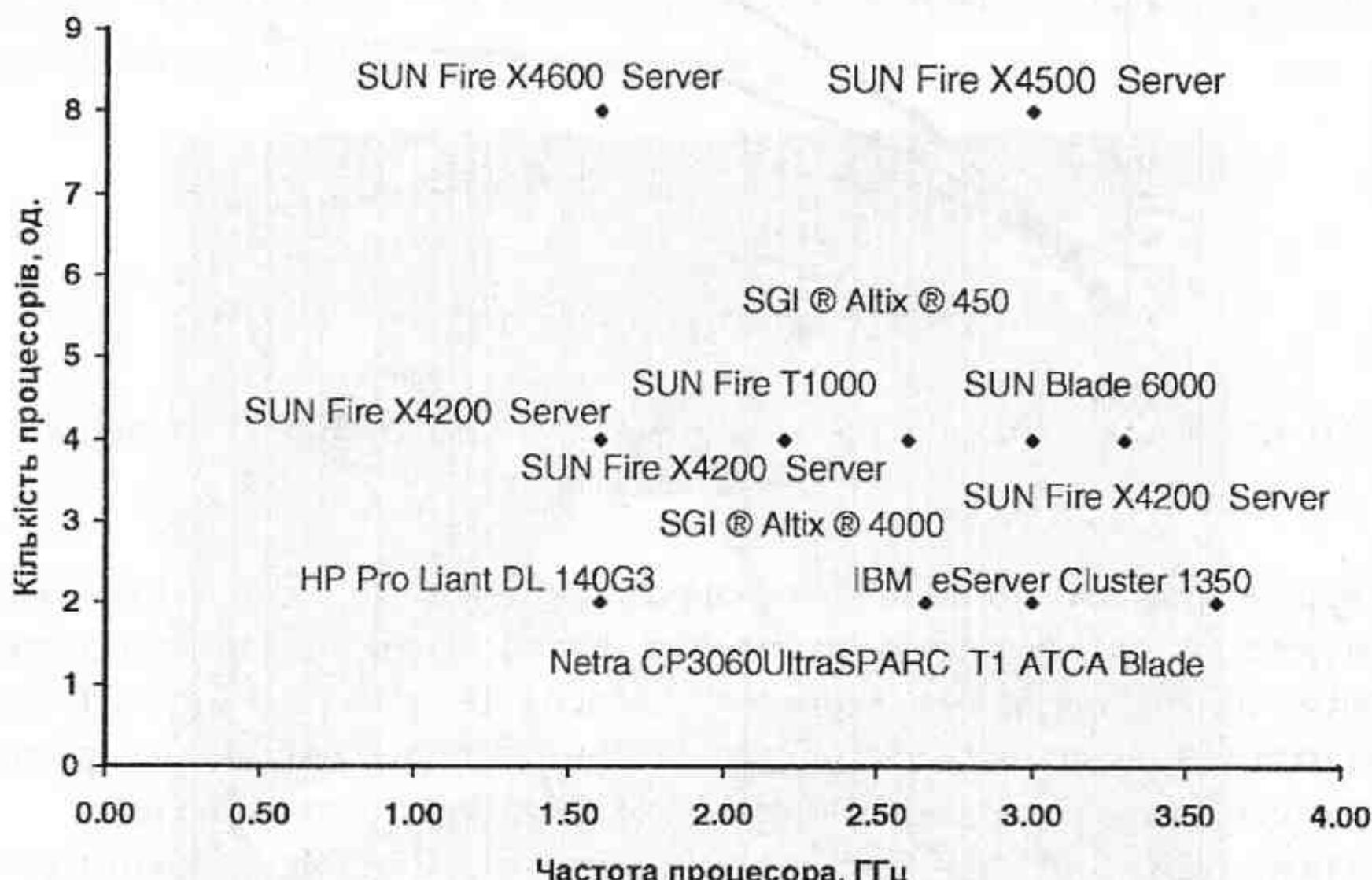


Рис. 12.4. Дані про реальні багатопроцесорні системи

Потрібно відзначити, що при створенні систем із масовою паралельною обробкою інформації, про які згадувалось вище, розробники орієнтуються не стільки на скорочення часу виконання алгоритмів, скільки на те, що відповідно до росту потужності системи будуть збільшуватись розміри вирішуваних задач.

12.3. Багатопотокова обробка інформації

Звичайно, перераховані вище підходи до підвищення продуктивності комп'ютерних систем, використовуючи паралельну обробку інформації, не виключають необхідність підвищення продуктивності однопроцесорних систем шляхом нарощування ресурсів процесора, наприклад, шляхом нарощування ємності кеш пам'яті та добавлення нових операційних пристрій. Хоча зрозуміло, що це тягне за собою зростання кількості транзисторів та площі кристалу, ускладнення процесора, і відповідно, вартості.

Іншим підходом є підвищення ефективності використання наявних ресурсів комп'ютерних систем з одним та декількома процесорами. Існує декілька дешевших, ніж масове розпаралелювання варіантів, які також дозволяють прискорити обчислення, а саме:

- Багатопроцесорність на одному кристалі (Chip Multiprocessing). Два процесорні ядра фізично розташовані на одному кристалі із використанням спільної або розподіленої кеш пам'яті. Природно, що розмір кристала отримують досить великим, що є причиною великої вартості цього кристала. В багатопроцесорній системі можуть функціонувати декілька таких кристалів.

- Багатопотокова обробка з квантуванням часу (Time-Slice Multithreading). Процесор перемикають поміж програмними потоками через фіксовані проміжки часу. Накладні витрати часом виходять досить значними, особливо якщо який-небудь процес знаходиться в стані очікування
- Багатопотокова обробка з перемиканням за подіями (Switch-on-Event Multithreading). Перемикання задач при наявності тривалих пауз, наприклад “невлучень” до кеш пам'яті, велике число яких притаманне серверним аплікаціям. Тут процес, що чекає на завантаження даних з порівняно повільної основної пам'яті до кеш пам'яті, пригальмовують, вивільняючи тим самим ресурси процесора на користь інших процесів. Проте багатопотокова обробка з перемиканням за подіями, як і багатопотокова обробка з квантуванням часу, не завжди дозволяє досягти оптимального використання ресурсів процесора, зокрема через помилки в передбаченні розгалужень, існуючі залежності поміж командами тощо

- Одночасна багатопотокова обробка (Simultaneous Multithreading). Тут програмні потоки виконуються на одному процесорі “одночасно”, тобто без переключення між ними. Ресурси процесора розподіляються динамічно, за принципом “не використовуєш – віддавай іншому”

Одночасна багатопотокова обробка покладена в основу технології гіперпотокової обробки (Hyper-Threading) фірми Intel, яку розглянемо детальніше

Багатопотокові обчислення використовуються не лише в серверах, де багатопотоківість існує первинно, але і в робочих станціях і настільних персональних комп'ютерах. Потоки можуть відноситися як до однієї, так і до різних прикладних програм, але майже завжди активних потоків більше від одного (аби переконатися в цьому, достатньо у Windows 2000/XP відчинити Task Manager та увімкнути відбиття ним числа виконуваних потоків). Разом з тим відомо, що стандартний процесор може одночасно опрацьовувати лише один з декількох існуючих потоків, тому і змушений постійно перемикатися поміж цими потоками

Технологію гіперпотокової обробки реалізовано в процесорі Intel Xeon MP (Foster MP), на якому і відбувалося дослідження ефективності цієї технології. Процесор Xeon MP використовує притаманне Pentium 4 ядро Willamette, містить 256 КВ кеш пам'яті другого рівня L2, 512 КВ кеш пам'яті третього рівня L3, та підтримує функціонування в чотирипроцесорних конфігураціях. Підтримка технології гіперпотокової обробки також присутня у процесорі для робочих станцій Intel Xeon (ядро Prestonia, 512 КВ кеш пам'яті другого рівня L2), що поступив на ринок дещо раніше від процесора Xeon MP. Далі розглянемо можливості технології гіперпотокової обробки на прикладі самого процесора Intel Xeon.

На рис. 12.5 а показано завантаженість процесорів комп'ютерної системи, яка складається з двох суперскалярних процесорів. Кожен процесор може виконувати по 3 команди в одному циклі. Незадіяні (позначено білим) прямокутники свідчать про неоптимальну утилізацію системних ресурсів. В той же час технологія гіперпотокової обробки не лише дозволяє опрацьовувати паралельно декілька потоків, але і зменшує при цьому наявне число незадіяних виконавчих часових інтервалів. На рис. 12.5 б показано завантаженість процесорів комп'ютерної системи, яка також складається з двох фізичних процесорів, та в якій використовується технологія гіперпотокової обробки. Кожен з процесорів тут опрацьовує по два потоки інформації, що дозволило наблизити систему до пікової продуктивності.

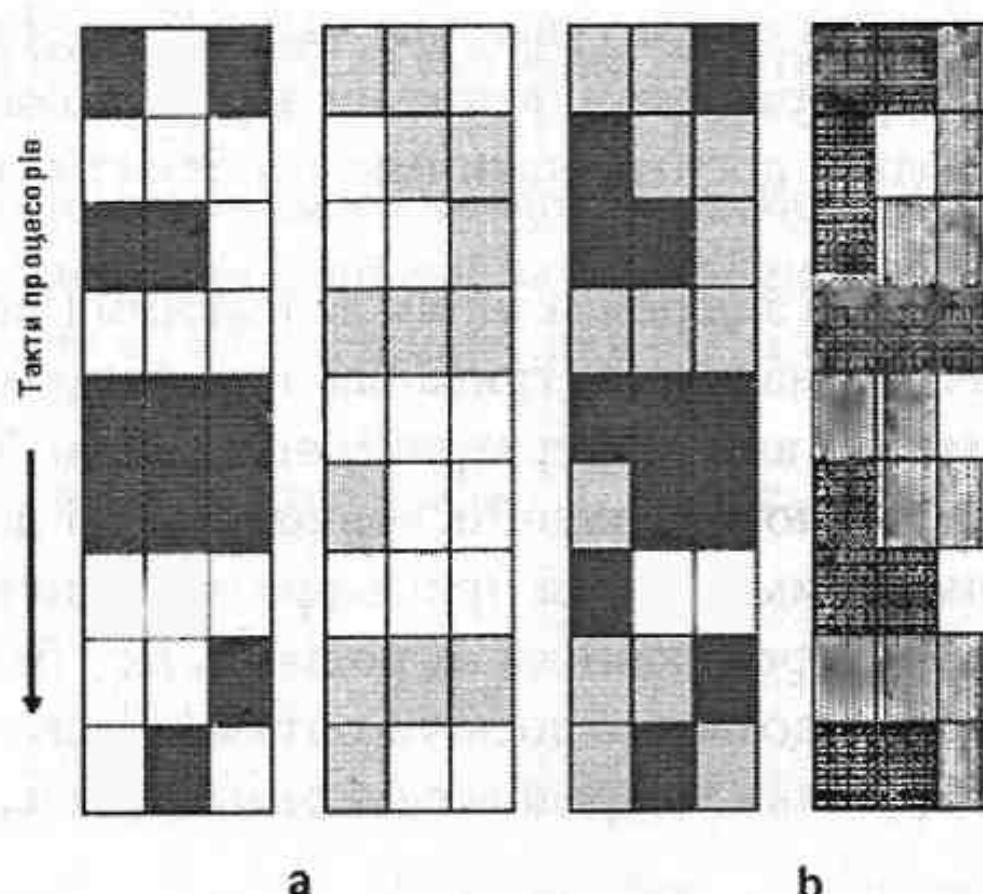


Рис. 12.5. Завантаженість процесорів комп’ютерної системи, яка складається з двох суперскалярних процесорів без використання технології гіперпотокової обробки (а) та з використанням технології гіперпотокової обробки (б)

Технологія гіперпотокової обробки ґрунтується на тому, що одночасно тільки частина ресурсів процесора використовується на опрацювання програмного коду. Не використувані ресурси можна завантажити іншою роботою, наприклад, задіяти на паралельне виконання ще одного додатку (або іншого потоку з того ж додатку). Заради цього в одному фізичному процесорі Intel Xeon формують два логічні процесори (LP – Logical Processor), що поділяють поміж собою обчислювальні ресурси єдиного фізичного процесора. Операційна система та прикладна програма “відчувають” саме два логічні процесори та спроможні розподіляти роботу між ними так само, як і у випадку повноцінної двопроцесорної системи. Поділ ресурсів (зокрема, виконавчих вузлів) поміж двома потоками подано на рис. 12.5б.

Потрібно відзначити, що технологією гіперпотокової обробки передбачено, аби при наявності лише одного активного потоку дозволити йому виконуватися з швидкодією, як і на звичайному процесорі, тобто щоб ефективність використання процесора не зменшувалася. Заради цього у процесорі передбачено два режими роботи: однозадачний (ОЗ) та багатозадачний (БЗ). У режимі ОЗ активним є лише один логічний процесор, який без обмежень користується наявними ресурсами. Інший логічний процесор призупинено командою HALT. Із появою другого програмного потоку логічний процесор, що не був задіяний, активується (за допомогою переривання поточного стану HALT). При цьому фізичний процесор перемикається до стану БЗ. Пригальмовування незадіяних логічних процесорів командою HALT покладено на операційну систему, яка, до речі, і відповідає за таке ж швидке виконання одного потоку, як і у випадку без використання технології Hyper-Threading.

Для кожного з двох логічних процесорів зберігають так званий архітектурний стан (Architecture State), який складено із станів регістрів різного типу – загального призначення, керуючих, регістрів контролера переривань і службових (рис. 12.6). У кожного логічного процесора є свій контролер переривань і множина регістрів, для коректної роботи з якими введена таблиця альтернативних назв регістрів, яка відслідковує від-

повідність поміж реєстрами загального призначення логічних процесорів та фізичного процесора (використовують одну таку таблицю на кожен логічний процесор).

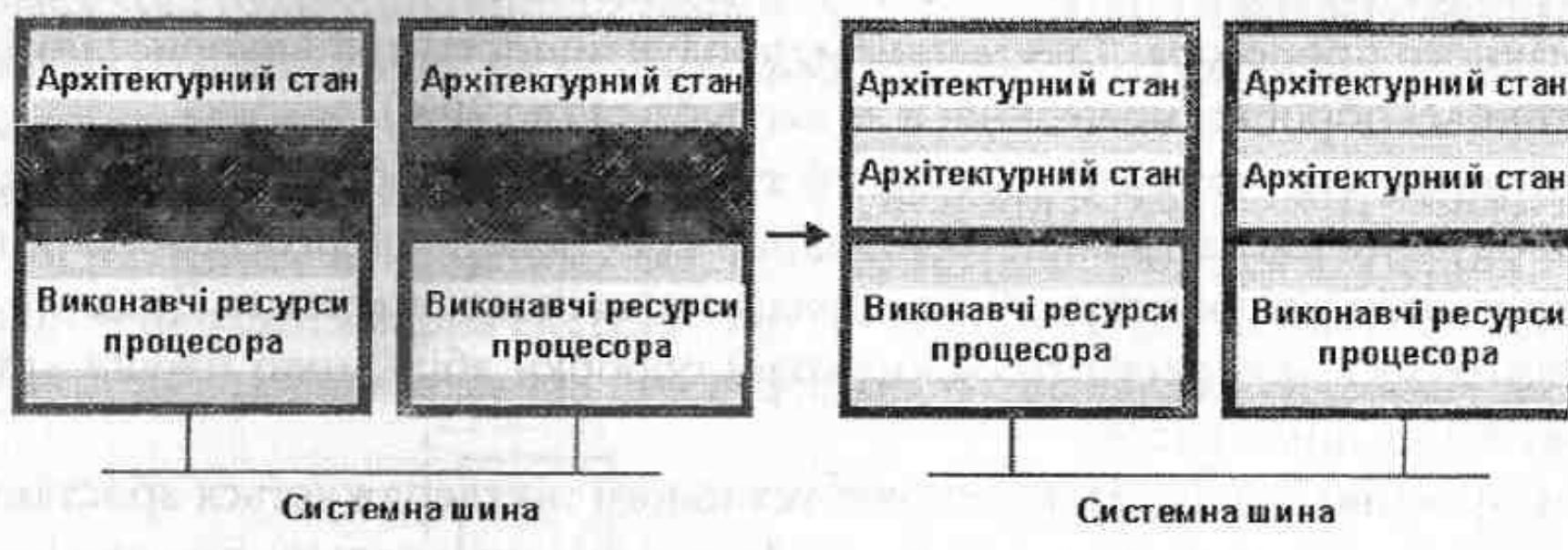


Рис. 12.6. Традиційна двопроцесорна система (a) та альтернативна система з підтримкою технології гіперпотокової обробки (b)

При паралельному опрацюванні двох потоків підтримується вміст двох лічильників команд. Переважна частка команд отримується з таблиці Trace Cache (TC), де ці команди зберігають в декодованому (трансьлованому до рівня мікрооперацій) виді. Доступ до TC обидва активні логічні процесори отримують за чергою, через такт. У той же час, коли активним є лише один логічний процесор, він отримує монопольний доступ до TC, без чергування за тактами. Так само відбувається і доступ до пам'яті команд, коли треба опрацювати складну команду, що не має динамічно компільованого варіанту для подання на безпосереднє виконання. Буфери перетворення з передісторією ITLB (Instruction Translation Look-aside Buffer), задіяні за умови відсутності необхідних команд у кеш пам'яті команд, дублюються і постачають команди за правилом "кожен для свого потоку". Блок декодування команд є поділюваним і у випадку, коли потрібне декодування команд для обох потоків, обслуговує їх за чергою (знову-таки через такт). Блоки черг мікрооперацій та їх розміщення поділено навпіл, аби розподілити елементи для кожного логічного процесора. Блоки диспетчерів у кількості 5 штук опрацьовують черги декодованих команд, незважаючи на приналежність до LP0 або LP1, і спрямовують команди на виконання цільовим виконавчим блокам залежно від готовності до виконання операндів команд і наявності вільного стану цільових виконавчих блоків (динамічне виконання). Кеш пам'яті усіх рівнів (L1/L2 для Xeon, а також L3 для Xeon MP) є цілком поділюваним поміж двома логічними процесорами, проте для забезпечення когерентності (цілісності) даних записи до буферів перетворення з передісторією DTLB (Data Translation Look-aside Buffer) забезпечуються дескрипторами у вигляді ідентифікаторів логічних процесорів.

Таким чином, команди обох логічних процесорів можна виконувати одночасно на ресурсах одного фізичного процесора, причому ці ресурси поділено на чотири наступні класи: дубльовані, цілком поділювані, із дескрипторами елементів, динамічно поділювані залежно від режиму роботи: однозадачний режим першого або другого логічного процесора чи багатозадачний режим.

Більшість прикладних програм, що отримують прискорення у багатопроцесорних системах, можуть також прискорюватися і на процесорі із увімкнутим режимом гіпер-

потокової обробки без будь-яких змін власних машинних кодів. Але існують і проблеми. Наприклад, якщо один процес знаходиться в стані очікування, то він спроможний захопити усі ресурси фізичного процесора, перешкоджаючи тим самим функціонуванню другого логічного процесора. Таким чином, продуктивність при використанні технології гіперпотокової обробки може іноді й знижуватися (до 20%).

Ще однією метою реалізації технології гіперпотокової обробки було зведення до мінімуму зростання числа транзисторів, площин кристала та енергоспоживання при постійному зростанні продуктивності. І це завдання вдалося виконати. Додання до Xeon/Xeon-MP підтримки технології гіперпотокової обробки збільшило площину кристала та енергоспоживання лише на 5%.

Загалом ефективність багатопотокової технології підтверджується зростанням продуктивності. Наступним рисунком (рис. 12.7) подано порівняльну продуктивність, що забезпечує двопроцесорна система без використання багатопотокової технології та із використанням цієї технології.



Рис.12.7. Продуктивність двопроцесорної системи залежно від кількості процесорів та залежно від використання (невикористання) технології гіперпотокової обробки (процесор Intel® Xeon™)

Бачимо зростання продуктивності (темний колір) при використанні другого процесора в порівнянні із базовим її рівнем для одного задіянного процесора, та додатковий пристрій продуктивності за рахунок увімкнення багатопотокової обробки (світлий колір).

12.4. Класифікація паралельних комп’ютерних систем

12.4.1. Класифікація Шора

Одна з перших класифікацій комп’ютерних систем була запропонована Д. Шором на початку 70-х років. Вона цікава тим, що є спробою виділення типових способів компонування комп’ютерних систем на основі фіксованого числа базових блоків: пристроя керування, арифметико-логічного пристрою, пам’яті команд і пам’яті даних. Додатково передбачається, що вибірка з пам’яті даних може здійснюватися словами, тобто вибираються всі розряди одного слова, і/або бітовим шаром – по одному розряду з однієї і тієї ж позиції кожного слова (іноді ці два способи називають горизонтальною і вертикальною

вибірками відповідно). Звичайно ж, при аналізі даної класифікації треба робити знижку на час її появи, оскільки передбачити велику різноманітність паралельних систем теперішнього часу тоді було у принципі неможливо. Отже, згідно з класифікацією Д. Шора, всі комп'ютери розбиваються на шість класів, перший з яких дістав назву машини I, другий – машини II, і т. д.

На поданих нижче рисунках позначено: ПІ – пам'ять команд (інструкцій), ПК – пристрій керування, АЛП – арифметико-логічний пристрій, ПД – пам'ять даних.

Машина I – це комп'ютерна система, яка містить пристрій керування, арифметико-логічний пристрій, пам'ять команд і пам'ять даних із послівною вибіркою (рис. 12.8).

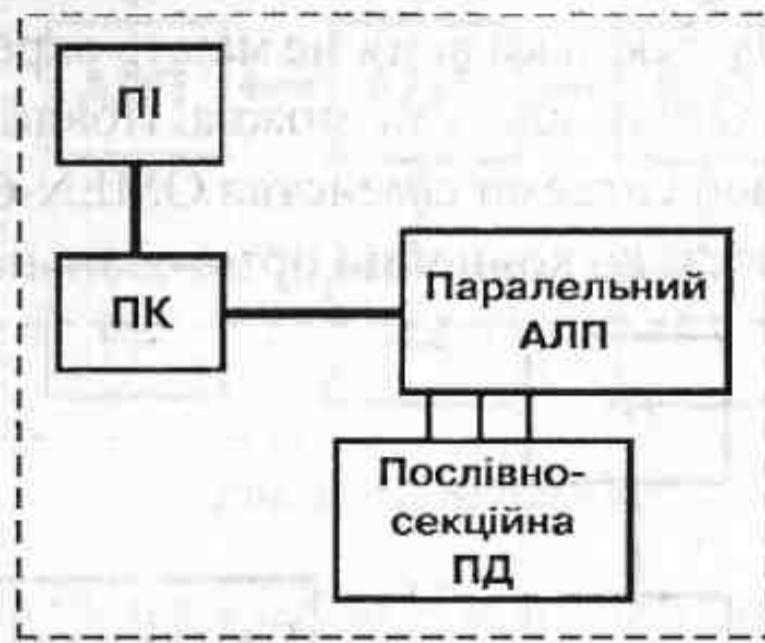


Рис. 12.8. Машина I

Зчитування даних здійснюється вибіркою всіх розрядів деякого слова для їх паралельної обробки в арифметико-логічному пристрої. Склад АЛП спеціально не вказується, що допускає наявність декількох функціональних пристрой, в тому числі конвеєрного типу. За цими міркуваннями до даного класу потрапляють як класичні послідовні машини (IBM 701, PDP-11, VAX 11/780), так і конвеєрні скалярні (CDC 7600) і векторно-конвеєрні (CRAY-1).

Якщо в машині I здійснювати вибірку не по словах, а по одному розряду з усіх слів, то одержимо машину II (рис. 12.9). Слови в пам'яті даних, як і раніше, розташовуються горизонтально, але доступ до них здійснюється інакше. Якщо в машині I відбувається послідовна обробка слів при паралельній обробці розрядів, то в машині II – послідовна обробка бітових шарів при паралельній обробці множини слів.

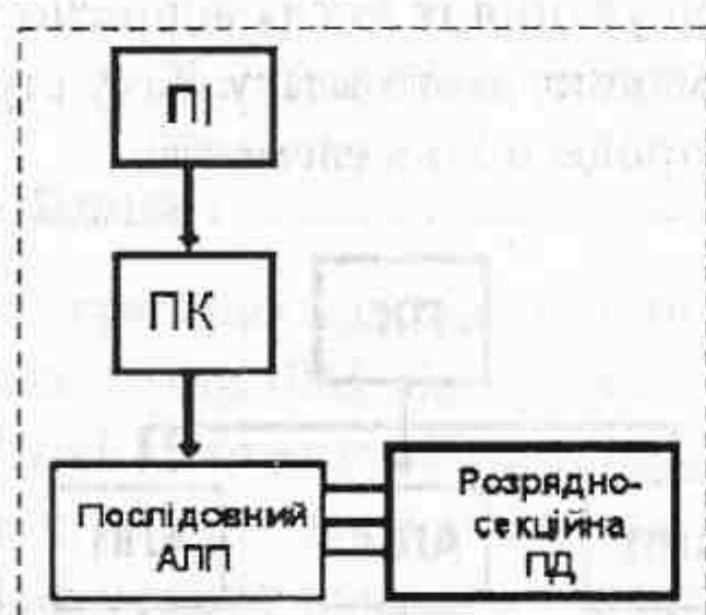


Рис. 12.9. Машина II

Структура машини II лежить в основі асоціативних комп'ютерів (наприклад, так побудовано центральний процесор машини STARAN), причому фактично такі комп'ютери мають в складі арифметико-логічного пристроя множину порівняно простих операцій-

них пристрів порозрядної обробки даних. Іншим прикладом служить матрична система ICL DAP, яка може одночасно обробляти по одному розряду з 4096 слів.

Якщо об'єднати принципи побудови машин I і II, то одержимо машину III (рис. 12.10). Ця машина має два арифметико-логічні пристрої – горизонтальний і вертикальний, а також модифіковану пам'ять даних, яка забезпечує доступ як до слів, так і до бітових шарів. Вперше ідею побудови таких систем у 1960 році висунув У. Шуман, що називав їх ортогональними (якщо пам'ять представляти як матрицю слів, то доступ до даних здійснюється в напрямі, "ортогональному" традиційному – не по словах (рядках), а по бітових шарах (стовпцях)). У принципі, як машину STARAN, так і ICL DAP можна запрограмувати на виконання функцій машини III, але оскільки вони не мають окремих АЛП для обробки слів і бітових шарів, віднести їх до даного класу не можна. Повноправними представниками машин класу III є обчислювальні системи сімейства OMEN-60 фірми Sanders Associates, побудовані в прямій відповідності до концепції ортогональної машини.

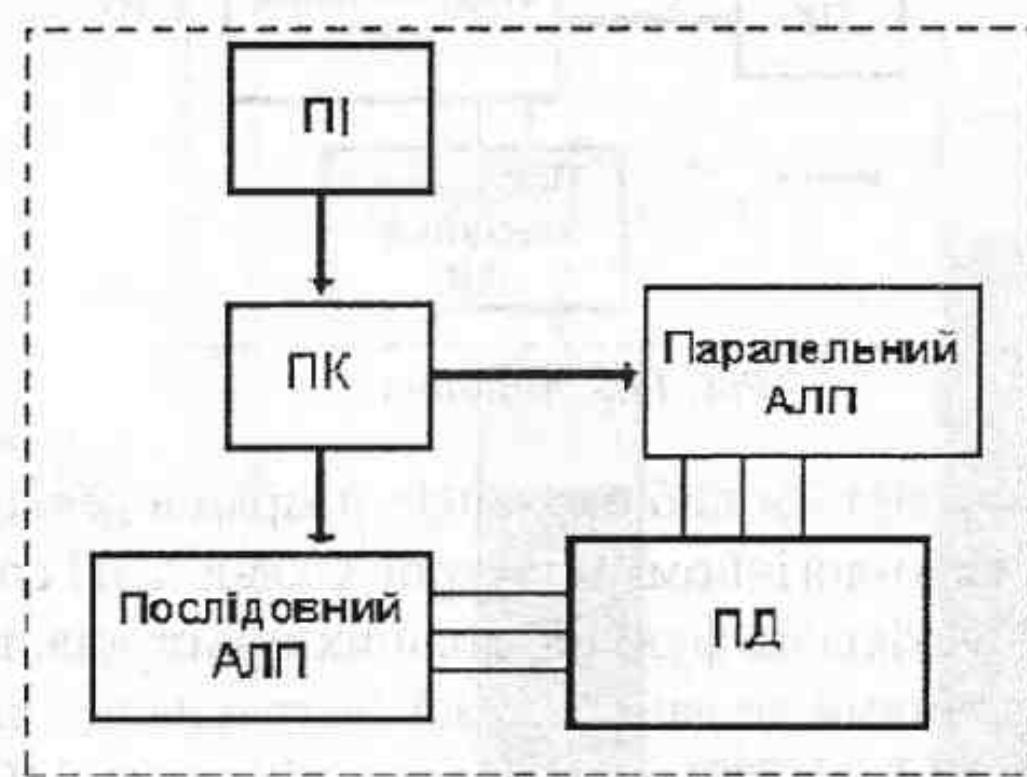


Рис. 12.10. Машина III

Якщо в машині I збільшити кількість пар арифметико-логічний пристрій <=> пам'ять даних (іноді цю пару називають процесорним елементом), то одержимо машину IV (рис. 12.11). Єдиний пристрій керування видає команду за командою відразу всім процесорним елементам. З одного боку, відсутність з'єднань між процесорними елементами робить подальше нарощування їх числа відносно простим, але з іншого боку, сильно обмежує застосовність машин цього класу. Таку структуру має обчислювальна система PEPE, яка об'єднує 288 процесорних елементів

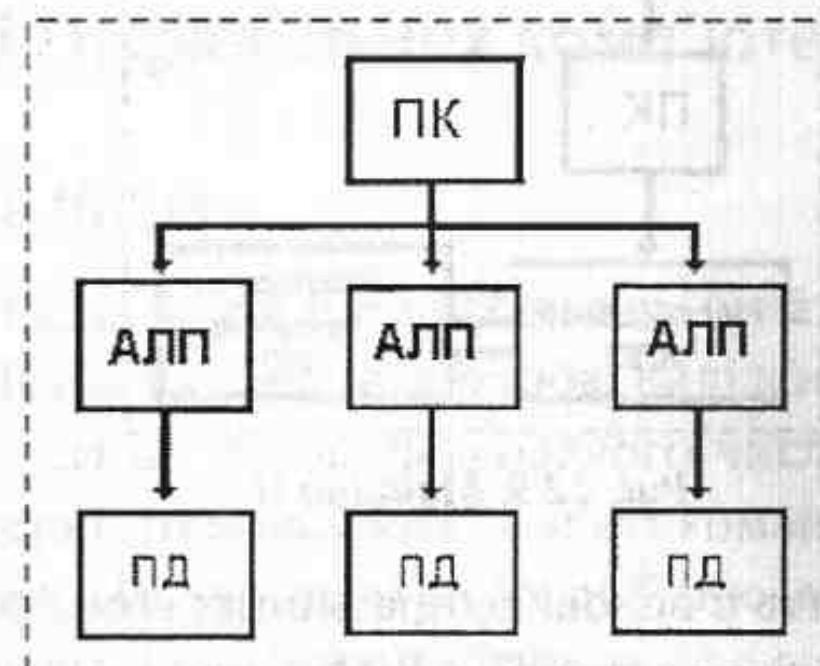


Рис. 12.11. Машина IV

Якщо ввести безпосередні лінійні зв'язки між сусідніми процесорними елементами машини IV, то одержимо структуру машини V (рис. 12.12). Будь-який процесорний елемент тепер може звертатися до даних як у своїй пам'яті, так і в пам'яті безпосередніх сусідів. Подібна структура характерна, наприклад, для класичної матричної багатопроцесорної системи ILLIAC IV.

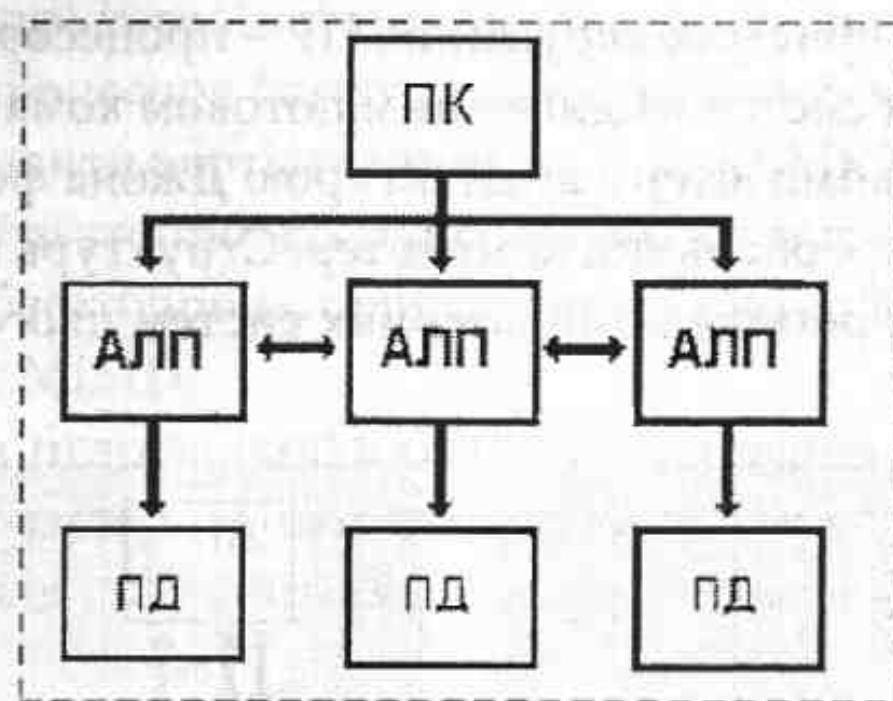


Рис. 12.12. Машина V

Відмітимо, що у всіх машинах з І-ї по V-у дотримана концепція розділення пам'яті даних і арифметико-логічних пристрой, припускаючи наявність шини даних або якого-небудь комутуючого елемента між ними. Машина VI, названа матрицею з функціональною пам'яттю (або пам'яттю з вбудованою логікою), є іншим підходом, що передбачає розподіл логіки процесора по всьому запам'ятовуючому пристрою (рис. 12.13). Прикладом можуть служити як прості асоціативні запам'ятовуючі пристрої, так і складні асоціативні процесори.

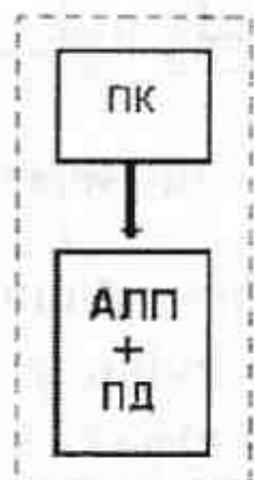


Рис. 12.13. Машина VI

12.4.2. Класифікація Фліна

Одну з перших практично значимих класифікацій паралельних комп'ютерних систем подав у 1966 році співробітник фірми IBM Майкл Флін, який зараз є професором Стенфордського університету (США). Його класифікація базується на оцінці потоку інформації, яка поділена на потік даних між основною пам'яттю та процесором, та потік команд, які виконує процесор. При цьому потік даних та команд може бути як одиничним, так і множинним. Згідно з М. Фліном, усі комп'ютерні системи поділяють наступним чином:

- ОКОД – комп'ютерні системи з одиничним потоком команд та одиничним потоком даних (SISD – Single Instruction stream Single Data stream).
- МКОД – комп'ютерні системи з множинним потоком команд та одиничним потоком даних (MISD – Multiply Instruction stream Single Data stream).

- ОКМД – комп’ютерні системи з одиничним потоком команд та множинним потоком даних (SIMD – Single Instruction stream Multiply Data stream).
- МКМД – комп’ютерні системи з множинним потоком команд та множинним потоком даних (MIMD – Multiply Instruction stream Multiply Data stream).

Розглянемо запропоновану М. Фліном класифікацію детальніше. На поданих нижче рисунках позначено: ПК – пристрій керування, ПР – процесор, ПД – пам’ять даних.

До класу комп’ютерних систем з одиничним потоком команд та одиничним потоком даних належить, зокрема, комп’ютер з архітектурою Джона фон Неймана, яким, наприклад, є розповсюджений персональний комп’ютер. Структура цієї системи представлена на рис. 12.14а. Організація роботи комп’ютерних систем цього класу була розглянута в попередніх розділах книги.

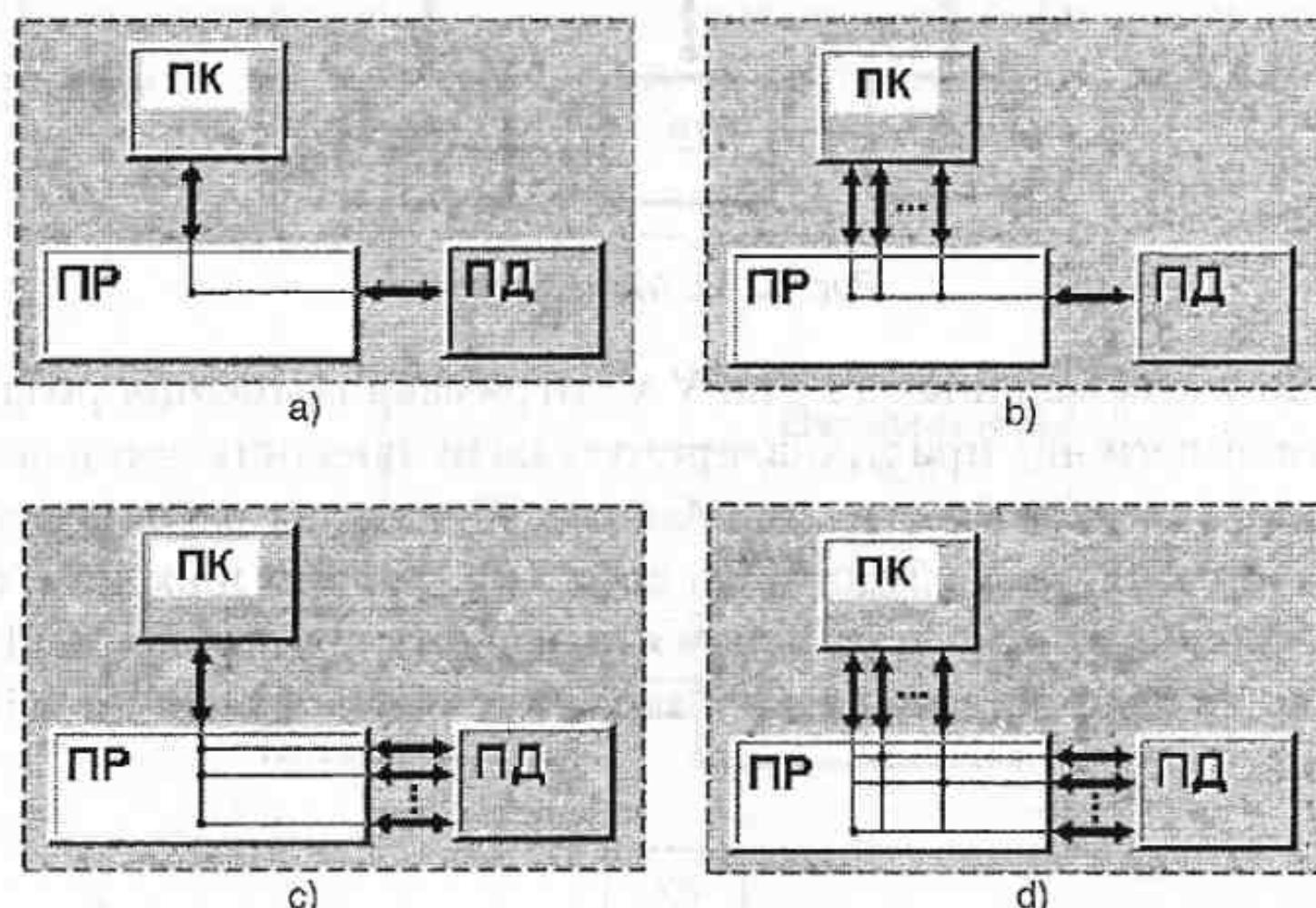


Рис. 12.14. Структура комп’ютерної системи типу ОКОД а), МКОД б), ОКМД с), МКМД д)

Структура комп’ютерної системи з множинним потоком команд та одиничним потоком даних показана на рис. 12.14б. Комерційні універсальні комп’ютерні системи цього типу на даний час невідомі, проте вони можуть з’явитися у майбутньому. До цього типу систем з деякими умовностями можна віднести спеціалізовані потокові процесори, зокрема систолічні, які використовують, наприклад, при обробці зображень.

В комп’ютерній системі з одиничним потоком команд та множинним потоком даних (рис. 12.14 с) одночасно обробляється велика кількість даних. До цього класу, зокрема, належать раніше розглянуті векторні процесори. До комп’ютерних систем з одиничним потоком команд та множинним потоком даних можна віднести також апаратну підсистему процесорів Pentium, яка реалізовує технологію MMX опрацювання даних для графічної операційної системи Windows.

Характерним прикладом комп’ютерної системи з одиничним потоком команд та множинним потоком даних може служити система, яка складається з двох частин: зовнішнього комп’ютера з архітектурою Джона фон Неймана, який виконує роль пристроя керування, і масиву ідентичних синхронізованих елементарних процесорів, здатних одночасно виконувати ту ж саму дію над різними даними. Кожен процесор у масиві має місцеву пам’ять невеликої ємності, де зберігаються дані, які обробляються паралельно.

З масивом процесорів з'єднано шину пам'яті зовнішнього комп'ютера таким чином, що він може довільно звернутися до кожного процесора масиву. Програма може виконуватися традиційно послідовно на зовнішньому комп'ютері, а її частина може паралельно виконуватися на масиві процесорів.

У комп'ютерній системі з множинним потоком команд та множинним потоком даних кожен процесор оперує із своїм потоком команд та своїм потоком даних (рис.12. – 4d). Як правило, окрім процесорів багатопроцесорної системи є серійними пристроями, що дозволяє значно зменшити вартість проекту. У класі МКМД треба відрізняти сильно зв'язані системи, власне багатопроцесорні системи, від мереж комп'ютерів, тобто слабо зв'язаних систем; тобто багатопроцесорні системи та комп'ютерні мережі потрапляють до різних підкласів класу МІМД.

В 1978 році Д. Куком було запропоновано розширення класифікації Фліна. У своїй класифікації Д. Кук розділив потоки команд та даних на скалярні та векторні потоки. Комбінація цих потоків приводить в підсумку до 16 типів архітектури паралельних комп'ютерних систем.

12.5. Типи архітектур систем ОКМД

До комп'ютерних систем класу ОКМД належать різні типи систем, які під керуванням одиничного потоку команд обробляють потоки даних. В першу чергу до цих систем потрібно віднести векторні та матричні комп'ютерні системи. Перші з них були детально розглянуті раніше. Призначення матричних комп'ютерних систем багато в чому схоже з призначенням векторних комп'ютерних систем – обробка великих масивів даних. В основі матричних комп'ютерних систем лежить матричний процесор, що складається з регулярного масиву процесорних елементів (ПЕ). Організація систем подібного типу на перший погляд достатньо проста. Вони мають пристрій керування, що генерує потік команд і велике число ПЕ, що працюють паралельно і оброблюють кожен свій потік даних. Проте на практиці, щоб забезпечити достатню ефективність системи при вирішенні широкого кола завдань, необхідно організувати зв'язки між процесорними елементами так, щоб якнайповніше завантажити їх роботою. Саме характер зв'язків між ПЕ і визначає різницю у властивостях системи.

Між матричними і векторними комп'ютерними системами є істотна різниця.

В складі системи команд векторного процесора є команди обробки векторів даних, що дозволяє ефективно завантажити конвеєри його операційних блоків. Векторні процесори простіше використовувати, тому що команди для обробки векторів – це зручна модель програмування, ніж команди для паралельно включених ПЕ.

Матричний процесор інтегрує безліч ідентичних процесорних елементів, логічно об'єднаних у матрицю і працюючих в стилі ОКМД. Не так істотно, як конструктивно реалізована матриця процесорних елементів – на одному кристалі чи на декількох. Важливий сам принцип – ПЕ логічно скомпоновані в матрицю і працюють синхронно, тобто присутній тільки один потік команд для всіх.

Є дві головні конфігурації, які були використані в матричних комп'ютерних системах класу ОКМД. В першій схемі кожен процесор Р має власну локальну пам'ять М (рис. 12.15).

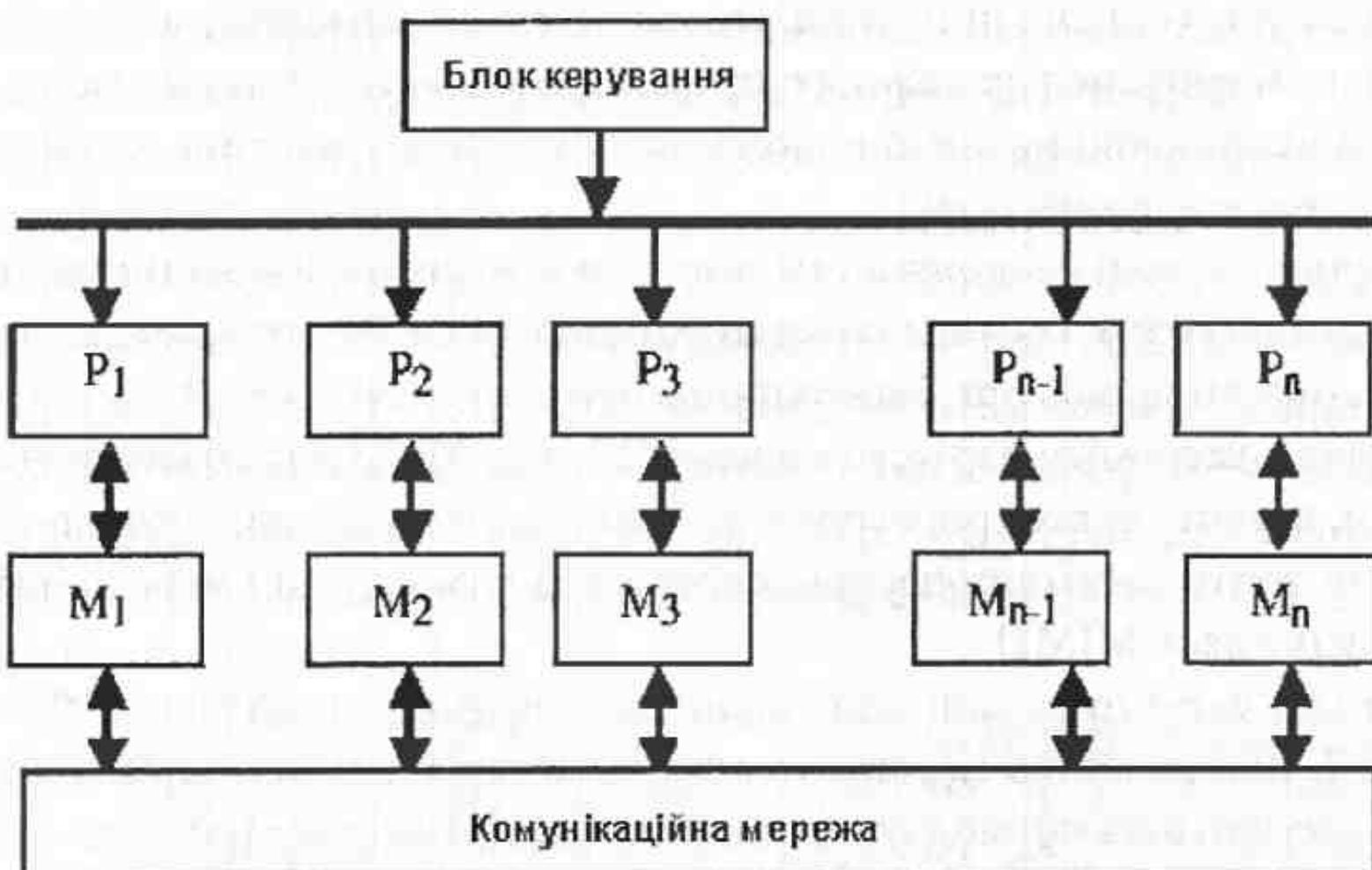


Рис. 12.15. Багатопроцесорна комп’ютерна система з одиничним потоком команд та множинним потоком даних із власною пам’яттю в кожного процесора

Процесори можуть зв’язатися один з одним через комунікаційну мережу. Якщо комунікаційна мережа не забезпечує прямого зв’язку між заданою парою процесорів, то згодом ця пара може обмінятися даними через проміжний процесор. Таку схему зв’язку було використано в комп’ютері ILLIAC IV. Комунікаційна мережа в ILLIAC IV дозволяла кожному процесору зв’язуватися безпосередньо з чотирма сусідніми процесорами. В матриці із 8x8 процесорів кожний i-й процесор міг контактувати безпосередньо з (i-1)-м, (i+1)-м, (i-8)-м, і (i+8)-м процесорами.

У другій схемі процесори і модулі пам’яті зв’язуються між собою через комунікаційну мережу (рис. 12.16). Два процесори можуть передати дані один одному через проміжний модуль пам’яті або, можливо, через проміжний процесор. За такою схемою, наприклад, побудовано процесор BSP фірми Burroughs.



Рис. 12.16. Багатопроцесорна комп’ютерна система з одиничним потоком команд та множинним потоком даних із спільною пам’яттю

12.6. Типи архітектур систем МКМД

Комп'ютерні системи класу МКМД складаються з багатьох процесорів та багатьох модулів пам'яті, з'єднаних за допомогою комунікаційної мережі. Вони можуть бути поділені на дві великі групи: зі спільною пам'яттю та з передачею повідомлень. На рис. 12.17 а та б показано загальну структуру цих двох груп комп'ютерних систем.

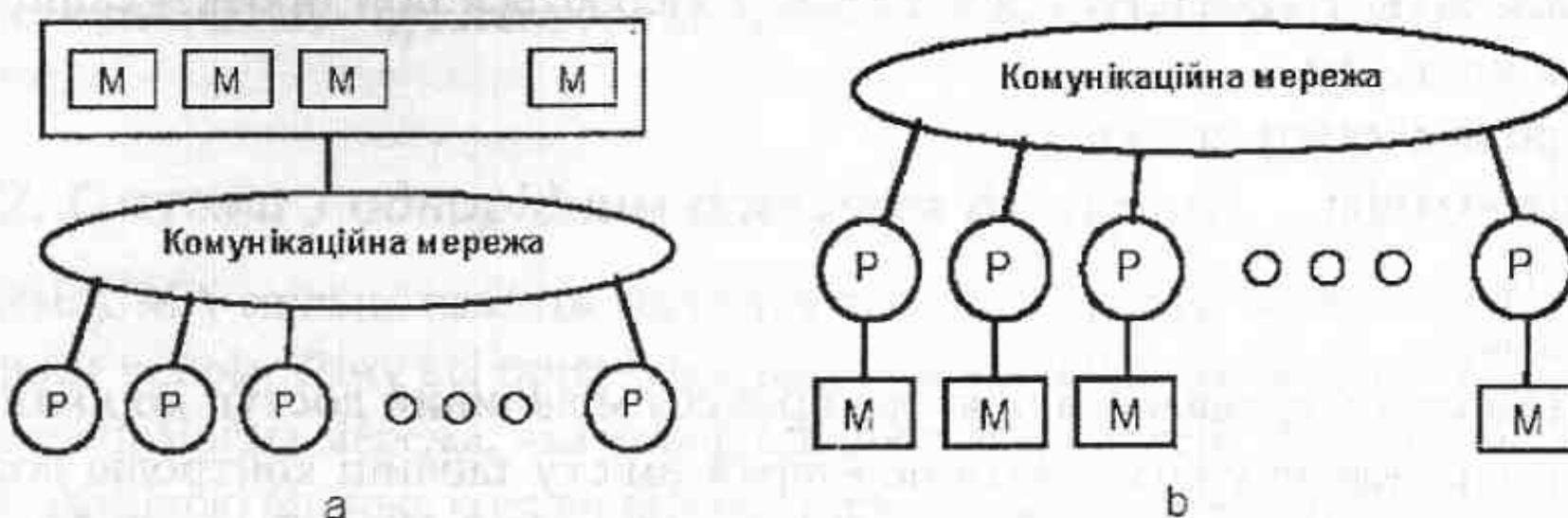


Рис. 12.17. Загальна структура комп'ютерних систем зі спільною пам'яттю а та з передачею повідомлень б

В першій групі процесори обмінюються інформацією через їх спільну пам'ять, причому кожний процесор має рівні можливості читати та записувати дані до пам'яті, а також однакову швидкість доступу до пам'яті, тому їх часто називають симетричними багатопроцесорними системами. Комерційними прикладами комп'ютерних систем першої групи є багатопроцесорні сервери фірм Sequent Computer's Balance and Symmetry, Sun Microsystems та Silicon Graphics Inc.

У другій групі процесори обмінюються інформацією через комунікаційну мережу. В комп'ютерній системі з передачею повідомлень (також їх називають системами з розподіленою пам'яттю) зазвичай наявна локальна пам'ять і процесор у кожному вузлі комунікаційної мережі. Тут відсутня спільна пам'ять, тому необхідно переміщувати дані з однієї локальної пам'яті до іншої за допомогою механізму передачі повідомлень. Це, зазвичай, робиться шляхом посилання-отримання кількох команд, які повинні бути вписані в прикладне програмне забезпечення. Комерційними прикладами систем передачі повідомлень є системи nCUBE, iPSC/2 і різні системи, базовані на трансп'ютерах. Ці системи кінець кінцем поступилися системам глобальної мережі Internet, в якій вузли є або серверами, або персональними комп'ютерами.

Архітектуру з розподіленою пам'яттю довелося використовувати із-за переходу до все більших систем. Потрібно відзначити, що програмування в системі зі спільною пам'яттю є простішим, а в системах передачі повідомлень забезпечується масштабованість. Тому з'явились комбіновані системи з розподіленою та з спільною пам'яттю, такі як SGI Origin2000, та інші.

12.7. Організація комп'ютерних систем із спільною пам'яттю

12.7.1. Типи комп'ютерних систем із спільною пам'яттю

При проектуванні багатопроцесорних систем ключовим є питання поділу даних між процесорами. Ідеальною відповіддю на це питання є застосування єдиного адресного

простору. Тут процесори взаємодіють через спільні змінні, що зберігаються в єдиній пам'яті з єдиним адресним простором для усіх процесорів. У системі із спільною пам'яттю процесори взаємодіють між собою шляхом зчитування-запису інформації з комірок спільної пам'яті, яка є однаково доступною для всіх процесорів. Кожен процесор може мати реєстри, буфери, кеш та локальну пам'ять, як додаткові ресурси пам'яті. Потрібно брати до уваги кілька основних проблем, які з'являються при проектуванні систем із спільною пам'яттю. Це:

- контроль доступу до пам'яті;
- синхронізація;
- захист;
- безпека.

Контроль доступу визначає, для якого процесу можливий доступ до яких ресурсів. Моделі контролю доступу вимагають перевірки вмісту таблиці контролю доступу для кожного запиту доступу від процесорів до спільної пам'яті. Ця таблиця вміщує пропорці, які визначають законністьожної спроби доступу. Якщо є спроба доступу до ресурсів, то поки бажаний доступ не завершений, всі спроби доступу нехтуються, і нелегальні процеси блокуються. Запити від сумісних процесів можуть змінити вміст таблиці контролю доступу протягом їх виконання. Пропорці контролю доступу з правилами синхронізації визначають функціональність системи.

Механізми синхронізації обмежують час доступу від сумісних процесів до спільних ресурсів. Досконала синхронізація гарантує, що інформація передається належним чином та забезпечується належна системна функціональність.

Захист – це системна особливість, що перешкоджає наданню процесам довільного доступу до ресурсів, що належать іншим процесам. Сумісне використання і захист є протилежними за функціями, оскільки сумісне використання дозволяє доступ, тоді як захист обмежує його.

Найпростіша система із спільною пам'яттю має один модуль пам'яті, який може бути доступний від двох процесорів (рис. 12.18). Запити надходять в модуль пам'яті через його два порти. Арбітражний блок у межах модуля пам'яті передає запити до диспетчера пам'яті. Якщо модуль пам'яті не зайнятий і надходить один запит, то арбітражний блок передає цей запит до диспетчера пам'яті і запит обслуговується. Модуль знаходиться в зайнятому стані під час обслуговування запиту. Якщо надходить новий запит, коли пам'ять зайнята, обслуговуючи попередній запит, процесор P, що послав запит, може утримувати свій запит на лінії, поки пам'ять не стає вільною, або може повторювати свій запит пізніше.

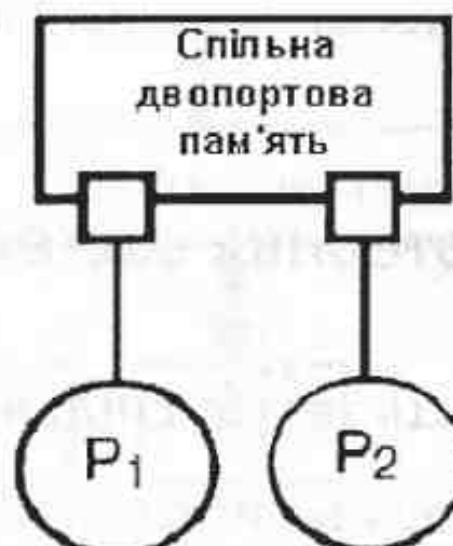


Рис. 12.18. Система із спільною пам'яттю з двома портами

Залежно від типу комунікаційної мережі, системи із спільною пам'яттю можна класифікувати наступним чином:

- системи UMA з однорідним доступом до пам'яті (UMA – uniform memory access);
- системи NUMA з неоднорідним доступом до пам'яті (NUMA – nonuniform memory access);
- системи COMA – архітектура лише з кеш-пам'яттю (COMA – cache-only memory architecture).

12.7.2. Системи з однорідним доступом до пам'яті

У системі UMA спільна пам'ять доступна всім процесорам через комунікаційну мережу одним чином. Тому всі процесори мають рівний час доступу до будь-якої комірки пам'яті. Комунікаційна мережа, яка використовується в системі UMA, може бути одничною або множинною шиною, координатною мережею чи багатопортовою пам'яттю.

Типова структура системи з однорідним доступом до пам'яті на основі одношинної комунікаційної мережі приведена на рис. 12.19 а.

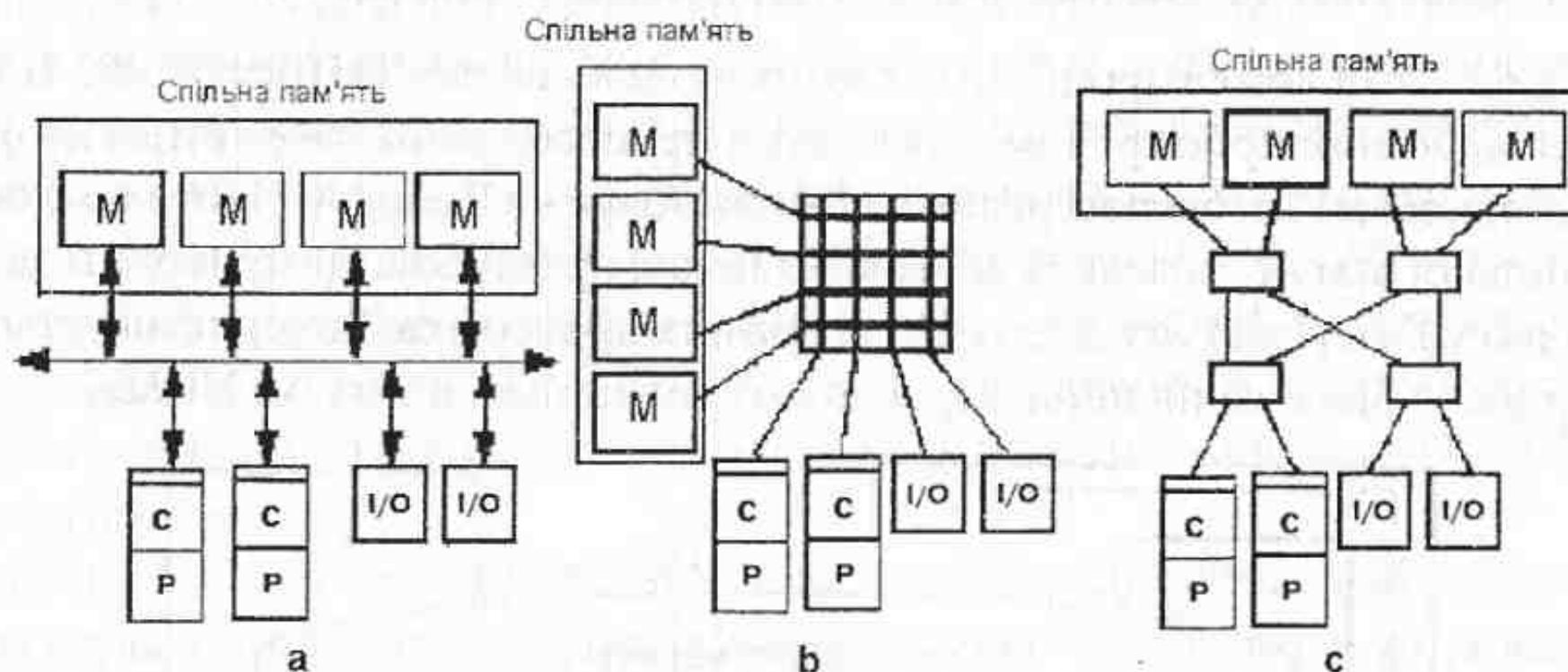


Рис. 12.19. Багатопроцесорні системи UMA із спільною пам'яттю на основі шинної топології а, координатного комутатора b та багатоярусної мережі с

В граничному випадку час пересилання через шину може бути зменшений до нуля після того, як вміст кеш пам'ятей C завантажено від спільної пам'яті. Ця організація пам'яті є найпопулярнішою серед систем із спільною пам'яттю. Вона дозволяє досить просто розширити систему шляхом підключення більшої кількості процесорів. Приклад цієї архітектури – сервери Sun Starfire servers, HP V series, і Compaq AlphaServer GS. Зрозуміло, що продуктивність цієї системи обмежена часом циклу шини. Тому кожен процесор має свою кеш пам'ять, що суттєво зменшує кількість звернень до шини. Наявність багатьох кеш пам'ятей породжує проблему їх когерентності, тобто несуперечності вмісту кеш пам'яті кожного процесора із вмістом спільної основної пам'яті багатопроцесорної системи. Зазначену проблему вирішують шляхом спостереження за шиною, що з'єднує процесори з пам'яттю, за допомогою контролера кожної кеш пам'яті разом із реалізацією в кожній кеш пам'яті наскрізного запису. Можна також в частині процесорів не використовувати кеш пам'ять.

Проблема когерентності пам'яті є однією з причин того, що багатопроцесорні системи із спільною пам'яттю на основі спільної шини мають невелику кількість процесорів.

Так, максимальна кількість процесорів типу Alpha 21264 в системі Compaq Alfa Server GS 140 рівна 14, процесорів PA-8500 в системі HP N9000 – 8, процесорів Power PC 604 у системі RS 6000 – 4.

Як вже вказувалось, використовувана в системі UMA комунікаційна мережа може бути одиничною чи множинною шиною, координатною мережею чи багатопортовою пам'яттю. На рис.12.19b та рис.12.19c подано типові логічні організації багатопроцесорних систем із спільною пам'яттю на основі топології координатної комутації та багатоярусної мережі. Тут буквою С позначено локальну кеш пам'ять, I/O – пристрой введення/виведення, Р – процесори, М – модулі спільної пам'яті.

Структури на основі топології координатної комутації (рис.12.19a) та багатоярусної мережі (рис.12.19b) дозволяють вирішити проблему обмеженої пропускної здатності системи із спільною шиною. Координатний комутатор та багатоярусна мережа забезпечують множинність шляхів з'єднання процесорів та блоків пам'яті. Тому тут кількість процесорів є зазвичай більшою, ніж в системах із спільною шиною.

12.7.3. Системи з неоднорідним доступом до пам'яті

У системі NUMA кожен процесор має частину спільної пам'яті (рис. 12.20). Ця пам'ять має єдиний адресний простір. Тому, будь-який процесор може звернутися до будь-якої комірки спільної пам'яті безпосередньо, використовуючи її адресу. Проте час доступу до модулів спільної пам'яті залежить від їх відстані від процесора. Це приводить до різного (неоднорідного) часу доступу до пам'яті від різних процесорів. Використовується кілька архітектур для підключення процесорів до модулів пам'яті в системі NUMA.

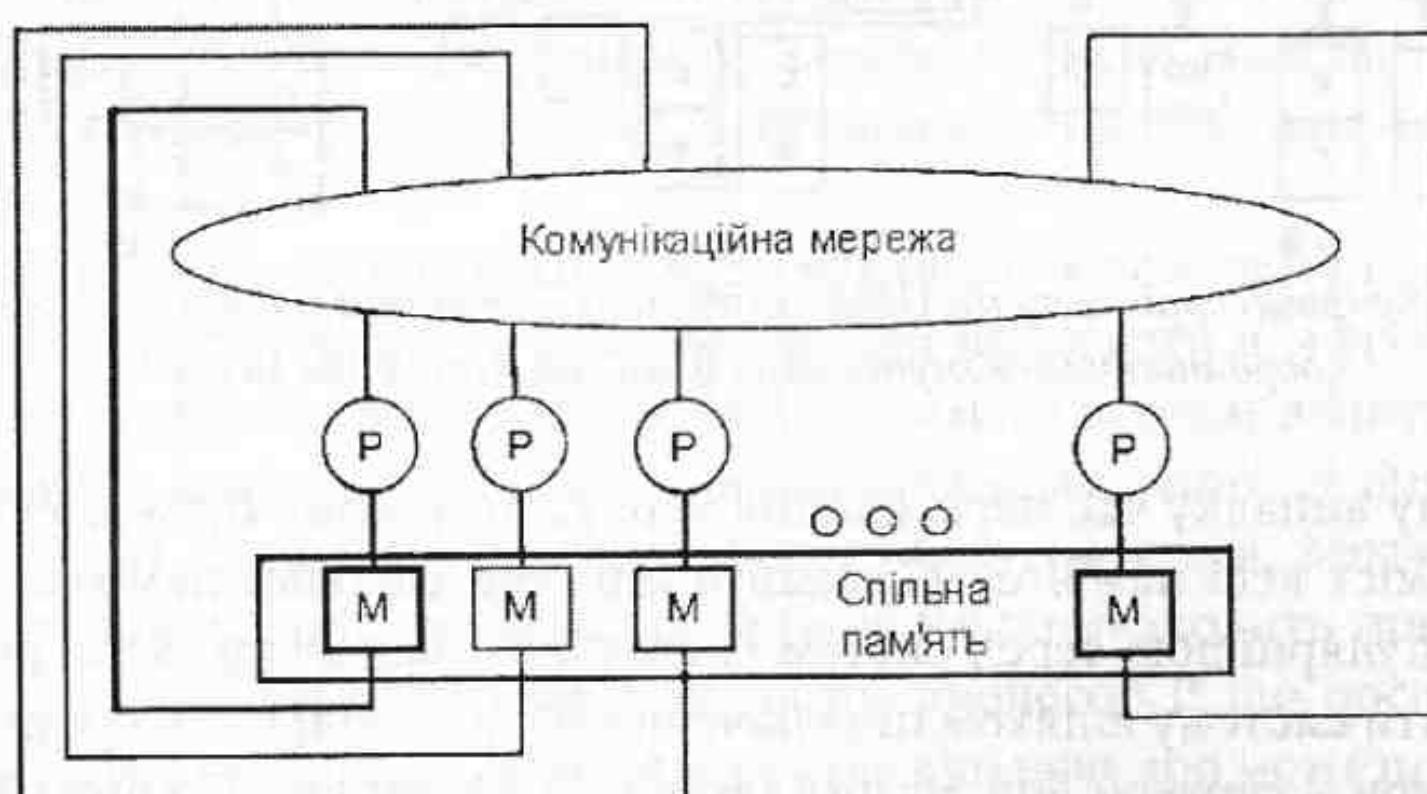


Рис. 12.20. Система NUMA із спільною пам'яттю

Як правило, система NUMA розрахована на застосування великої кількості процесорів. UMA є більш вибагливою щодо швидкодії підсистеми пам'яті. Тобто реалізація UMA є порівняно простішою та розповсюдженішою (багатопроцесорні сервери фірми Compaq, як приклад). Архітектуру NUMA реалізовано, наприклад, в суперкомп'ютері ORIGIN 2000 фірми SGI (більш точно – тут є архітектура CC-NUMA, або NUMA із множиною когерентних кеш пам'ятей).

У симетричних багатопроцесорних комп'ютерних системах із спільною пам'яттю має місце практична межа числа їх процесорів. Ефективна схема з кеш пам'яттю зменшує

навантаження на шину між процесором і основною пам'яттю, але зі збільшенням числа процесорів навантаження на шину також зростає. Оскільки шина використовується також для передачі сигналів, що забезпечують когерентність, ситуація з навантаженням на шину ще більш ускладнюється. З якогось моменту в плані продуктивності шина перетворюється на вузьке місце. Для систем із спільною пам'яттю такою межею стає число процесорів в межах від 16 до 64. Наприклад, кількість процесорів системи Silicon Graphics Power Challenge обмежена 64 процесорами типу R10000, оскільки при подальшому збільшенні числа процесорів продуктивність падає.

Обмеження на число процесорів в архітектурі з спільною пам'яттю служить спонукальним мотивом для розвитку кластерних систем. В останніх кожен вузол має локальну основну пам'ять, тобто додатки "не бачать" спільної основної пам'яті. По суті, когерентність підтримується не стільки апаратурою, скільки програмним забезпеченням, що не кращим чином позначається на продуктивності. Одним із шляхів створення великомасштабних комп'ютерних систем є технологія CC-NUMA. Наприклад, система NUMA Silicon Graphics Origin підтримує до 1024 процесорів R10000. Технологія CC-NUMA передбачає включення множини незалежних вузлів, кожний з яких може бути, наприклад, системою із спільною пам'яттю. Таким чином, вузол містить множину процесорів, у кожного з яких присутні локальні кеш пам'яті первого і другого рівнів. У вузлі є й основна пам'ять, спільна для всіх процесорів цього вузла, але така, що розглядається як частина спільної основної пам'яті системи. В архітектурі CC-NUMA вузол виступає основним будівельним блоком. Наприклад, кожен вузол у системі Silicon Graphics Origin містить два мікропроцесори MIPS R10000. Вузли об'єднуються за допомогою якої-небудь комунікаційної мережі, яка представлена координатним комутатором, кільцем або має іншу топологію.

Відповідно до технології CC-NUMA, кожен вузол у системі володіє власною основною пам'яттю, але з погляду процесорів має місце спільна пам'ять, де кожен елемент будь-якої локальної основної пам'яті має унікальну системну адресу. Коли процесор ініціює доступ до пам'яті і потрібна комірка відсутня в його локальній кеш пам'яті, кеш-пам'ять другого рівня процесора організує операцію вибірки. Якщо потрібна комірка знаходиться в локальній основній пам'яті, вибірка проводиться з використанням локальної шини. Якщо ж необхідна комірка зберігається у віддаленій секції спільної пам'яті, то автоматично формується запит, що посилається по комунікаційній мережі на потрібну локальнушину, і вже по ній до підключеної до даної локальної шини кеш пам'яті. Всі ці дії виконуються автоматично, прозорі для процесора і його кеш пам'яті.

У даній конфігурації головна турбота – когерентність кеш-пам'ятей. Хоча окремі реалізації і відрізняються в деталях, загальним є те, що кожен вузол має довідник, в якому зберігається інформація про місцезнаходження в системі кожної складової спільної пам'яті, а також про стан кеш пам'яті.

12.7.4. Системи лише з кеш пам'яттю

Подібно до NUMA в системі COMA кожен процесор має частину спільної пам'яті (рис. 12.21). Проте в даному випадку спільна пам'ять є кеш пам'яттю. Система COMA вимагає, щоб дані мігрували до процесора, що запросив їх.

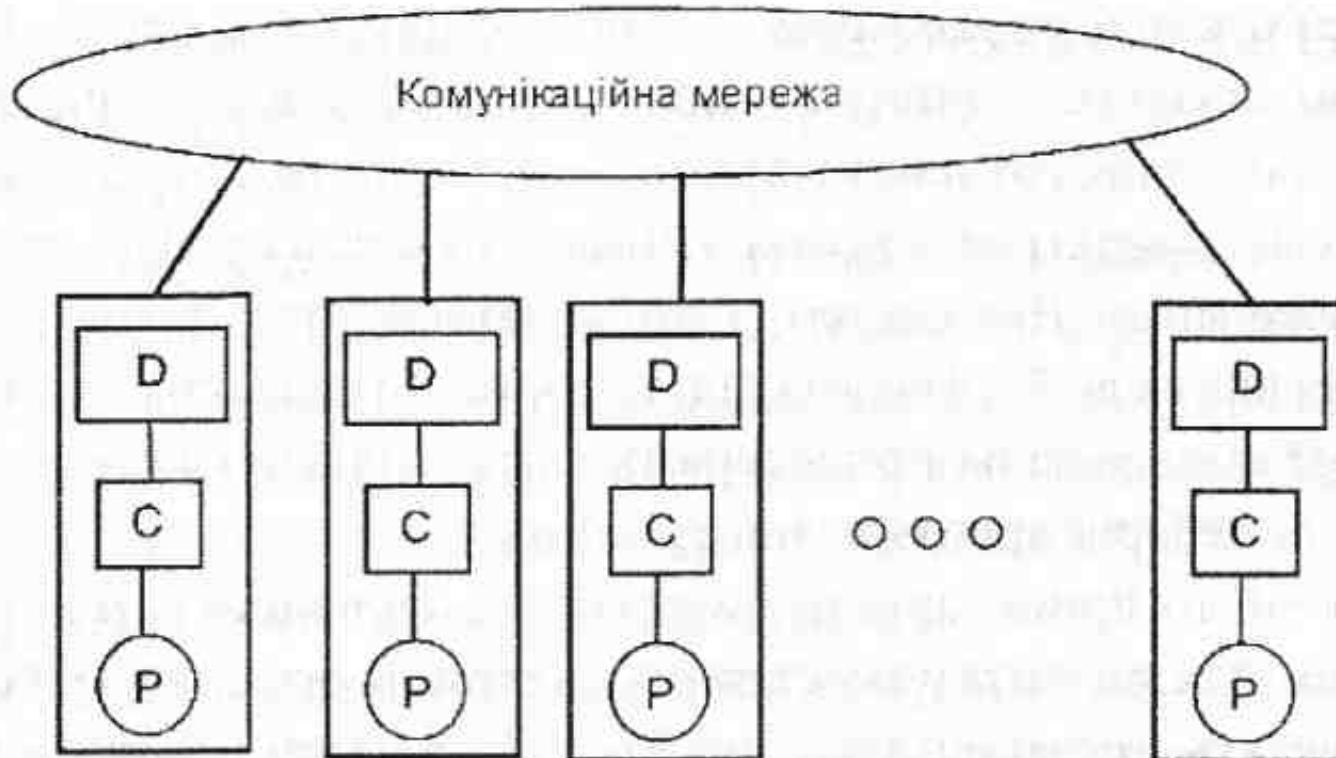


Рис. 12.21. Система СОМА зі спільною пам'яттю

Тут відсутня ієрархія пам'яті, а адресний простір визначається ємністю кеш пам'яті. Крім того, тут наявна директорія D кеш пам'яті, яка допомагає у віддаленому доступі до кеш пам'яті. Прикладом цієї архітектури є система KSR-1 фірми Kendall Square Research.

12.8. Організація комп'ютерних систем із розподіленою пам'яттю

Системи передачі повідомлень – клас багатопроцесорних систем, в яких кожен процесор має доступ до своєї локальної пам'яті. На відміну від систем із спільною пам'яттю, комунікації в системі передачі повідомлень виконуються шляхом посилання та отримання повідомлень. Вузол в такій системі складається з процесора і його локальної пам'яті (рис. 12.22).

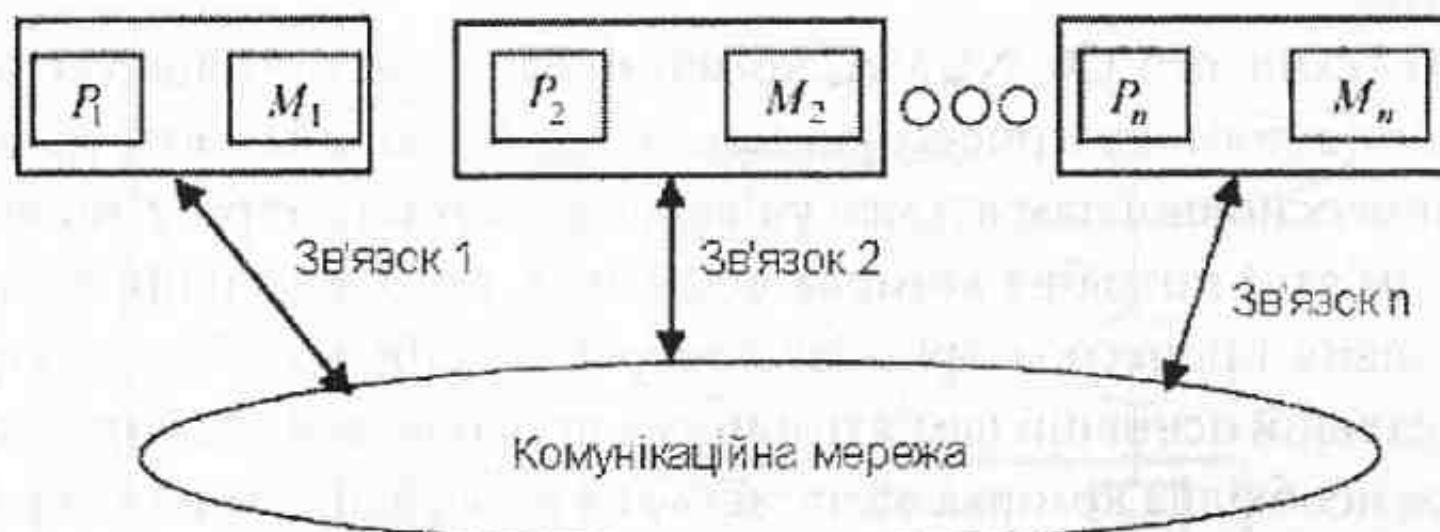


Рис. 12.22. Система з розподіленою пам'яттю

Вузли зазвичай запам'ятовують повідомлення в буферах (тимчасові комірки пам'яті, де повідомлення чекають до того часу, коли вони можуть бути послані або одержані), і виконують посилання-отримання повідомлення під час обробки даних. Паралельна обробка повідомлення і поточного обчислення здійснюється під керуванням операційної системи. Процесори не розділяють спільну пам'ять і кожен процесор має доступ до його власного адресного простору. Вузли в системі передачі повідомлень можуть бути з'єднані різноманітними способами – від спеціалізованих комутаторів до географічно розподілених мереж.

Підхід передачі повідомлень є масштабованим до великих пропорцій, тобто число процесорів може бути збільшено без істотного зменшення ефективності їх взаємодії.

Для організації зв'язку системи передачі повідомлень використовують статичні мережі, зокрема гіперкубічні мережі, які були досить популярні протягом багатьох років. Двовимірні і тривимірні решітчасті мережі так само широко використовувались в системах передачі повідомлень. Два важливі чинники повинні розглядатися в проектуванні комунікаційної мережі для систем передачі повідомлень. Це – пропускна здатність і мережний час очікування. Пропускна здатність визначена як кількість бітів інформації, переданих за одиницю часу (біт/сек). Мережний час очікування визначений як час, потрібний для передачі повідомлення. В 1987 році була введена блокова маршрутизація як альтернатива традиційній маршрутизації з проміжним зберіганням для того, щоб скоротити розмір необхідних буферів і щоб зменшити час очікування повідомлення. У блоковій маршрутизації пакет ділиться на менші блоки, які називають блоками керування потоком даних, так, що ці блоки рухаються конвеєрним методом разом з блоком заголовка до вузла призначення. Коли блок заголовка блокується через мережний затор, наступні за ним блоки блокуються також.

12.9. Комунікаційні мережі багатопроцесорних систем

12.9.1. Типи комунікаційних мереж

В основі архітектури будь-якої багатопроцесорної комп'ютерної системи лежить здатність до обміну даними між її компонентами. Це забезпечується комунікацією мережею (КММ), яка з'єднує між собою вузли комп'ютерної системи за допомогою каналів передачі даних (каналів зв'язку). В ролі вузлів можуть виступати процесори, модулі пам'яті, пристрой введення-виведення, комутатори або декілька перерахованих елементів, об'єднаних у функціональний пристрій. Організація внутрішніх комунікацій комп'ютерної системи називається топологією.

Топологію комунікаційної мережі визначає множина вузлів, які об'єднані множиною каналів. Зв'язок між вузлами зазвичай реалізується по двоточковій схемі (point-to-point). Будь-які два вузли, зв'язані каналом зв'язку, називають суміжними вузлами або сусідами. Кожен канал з'єднує один вузол-джерело з одним вузлом-приймачем. Канал характеризується кількістю сигнальних ліній, частотою або швидкістю передачі бітів по кожній сигнальній лінії, затримкою – часом пересилання біта з одного вузла до іншого. Для більшості каналів затримка знаходиться в прямій залежності від фізичної довжини лінії зв'язку та швидкості розповсюдження сигналу.

Вузол у мережі може бути термінальним, тобто джерелом або приймачем даних, комутатором, що пересилає інформацію з вхідного порту на вихідний, або суміщати обидві ролі. У мережах із прямиими зв'язками кожен вузол одночасно є як термінальним вузлом, так і комутатором, і повідомлення пересилаються між термінальними вузлами безпосередньо. У мережах з непрямими зв'язками вузол може бути або термінальним, або комутатором, але не одночасно, тому повідомлення передаються опосередковано, за допомогою виділених комутуючих вузлів. Існують також такі топології, які не можна однозначно зарахувати ні до прямих, ні до непрямих. Будь-яку пряму КММ можна зобразити у вигляді непрямої, розділивши кожен вузол на двотермінальний вузол і вузол комутації. Сучасні прямі мережі реалізуються саме таким чином – комутатор від-

діляється від термінального вузла і поміщається у виділений маршрутизатор. Основна перевага прямих КММ полягає в тому, що комутатор може використовувати ресурси термінальної частини свого вузла. Це стає істотним, якщо врахувати, що, як правило, останній включає комп'ютер або процесор.

Комуникаційні мережі багатопроцесорних систем можуть бути поділені на наступні групи:

- залежно від того, чи залишається конфігурація взаємозв'язків незмінною, при наймні поки виконується певне завдання, розрізняють мережі із статичною і динамічною топологіями;
- залежно від способу функціонування: синхронні та асинхронні;
- залежно від стратегії керування: централізовані та децентралізовані;
- залежно від способу перемикання: схемні та пакетні.

З'єднання в статичній мережі є фіксованими, тоді як в динамічній мережі вони можуть змінюватися в процесі роботи мережі за допомогою програмних засобів. При цьому статичні комунікаційні мережі в свою чергу ділять на одно-, двовимірні та гіперкубічні.

Динамічні комунікаційні мережі ділять на шинні (одно та багатошинні) та комутуючі (одноярусні, багатоярусні та координатні), як це показано на рис. 12.23.



Рис.12.23. Топології комунікаційних мереж

Дві можливі стратегії операцій взаємодії в мережі – це синхронна й асинхронна. У синхронних КММ всі дії жорстко узгоджені в часі, що забезпечується за рахунок єдиного генератора тактових імпульсів (ГТІ), сигнали якого одночасно транслюються у всі вузли. В асинхронних мережах єдиного генератора немає, а функції синхронізації розподілені по всій системі, причому в різних частинах мережі часто використовуються локальні ГТІ.

Залежно від вибраної стратегії комутації розрізняють мережі з комутацією з'єднань і мережі з комутацією пакетів. Як у першому, так і в другому варіанті інформація пересилається у вигляді пакету. Пакет є групою бітів, для позначення якої застосовують також термін повідомлення.

У мережах з комутацією з'єднань шляхом відповідного встановлення комутуючих елементів мережі формується канал від вузла-джерела до вузла-приймача, що зберіга-

ється, поки весь пакет не досягне пункту призначення. Пересилання повідомлень між певною парою вузлів проводиться завжди поодинці і за тим же маршрутом.

В мережі з комутацією пакетів приймається, що повідомлення самостійно знаходить свій шлях до місця призначення. На відміну від мереж із комутацією з'єднань, маршрут від початкового пункту до пункту призначення кожного разу може бути іншим. Пакет послідовно проходить через вузли мережі. Черговий вузол запам'ятовує прийнятий пакет у своєму буфері тимчасового зберігання, аналізує його і робить висновки, що з ним робити далі. Залежно від завантаженості мережі ухвалюється рішення про можливість негайногого пересилання пакету до наступного вузла і про подальший маршрут проходження пакету на шляху до пункту призначення. Якщо всі можливі канали для переміщення пакету до чергового вузла зайняті, в буфері вузла формується черга пакетів, яка "розсмоктується" у міру звільнення ліній зв'язку між вузлами (якщо черга також насичується, то відповідно до однієї із стратегій маршрутизації може відбутися так зване "скидання хвоста", тобто відмова від пакетів, що знову поступають).

КММ можна також класифікувати по тому, як в них організоване керування. У деяких мережах, особливо з комутацією з'єднань, прийняте централізоване керування (рис.12.24).

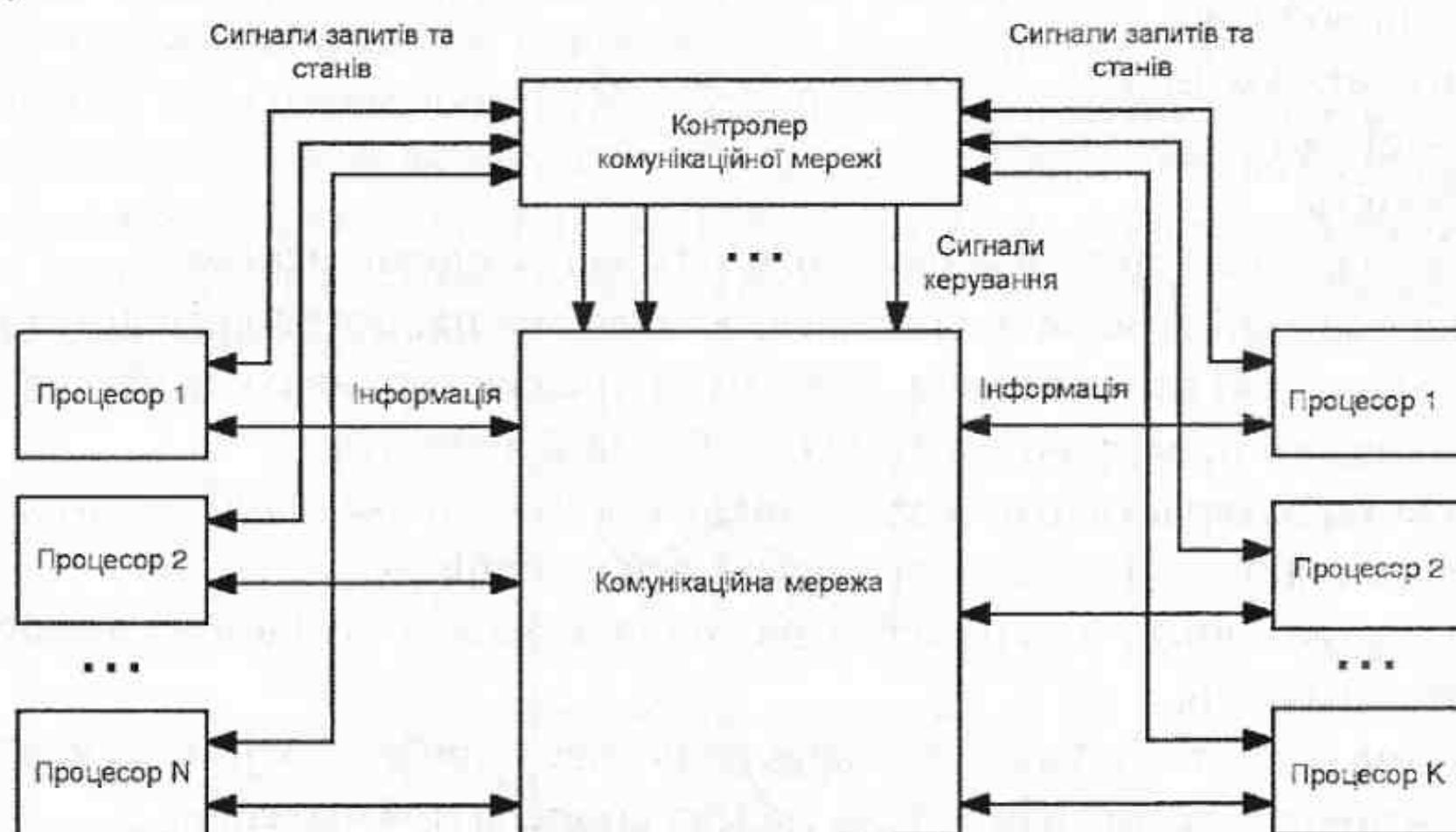


Рис.12.24. Багатопроцесорна система на основі комунікаційної мережі з централізованим керуванням

Процесори посилають запит на обслуговування в контролер комунікаційної мережі, який проводить арбітраж запитів із врахуванням заданих пріоритетів, і встановлює потрібний маршрут. До даного типу слід віднести мережі з шинною топологією. Процесорні матриці також будуються як мережі з централізованим керуванням, яке здійснюється сигналами від центрального процесора. Приведена схема застосовна і до мереж з комутацією пакетів. Тут тег маршрутизації, що зберігається в заголовку пакету, визначає адресу вузла призначення. Більшість серійних комп'ютерних систем має саме цей тип керування.

У схемах із децентралізованим керуванням функції керування розподілені по вузлах мережі. Варіант із централізованим керуванням простіше реалізується, але розширення мережі в цьому випадку пов'язане із значними труднощами. Децентралізовані мережі

в плані підключення додаткових вузлів є значно гнучкішими, проте взаємодія вузлів у таких мережах є істотно складнішою.

У ряді мереж зв'язок між вузлами забезпечується за допомогою деякої множини комутаторів, але існують також мережі з одним комутатором. Наявність великого числа комутаторів веде до збільшення часу передачі повідомлення, але дозволяє використовувати прості комутуючі елементи. Подібні мережі зазвичай будують як багатоярусні.

12.9.2. Основні характеристики комунікаційних мереж багатопроцесорних систем

До основних характеристики комунікаційних мереж багатопроцесорних систем належать наступні:

- розмір;
- складність;
- кількість зв'язків;
- діаметр;
- затримка;
- частота роботи;
- пропускна здатність;
- здатність до розширення;
- надійність.

Розмір мережі визначається кількістю вузлів, які об'єднані мережею.

Складність мережі вимірюється кількістю обладнання, потрібного для її реалізації. При цьому одиницею вимірювання може бути транзистор, вентиль або вузол деякої приведеної складності, наприклад, двовходовий мультиплексор.

Важливою характеристикою мережі є кількість зв'язків (каналів) між вузлами мережі, яка суттєво впливає на складність мережі та її надійність.

Діаметр мережі – число вузлів, через які повинне пройти повідомлення, щоб досягти його місця призначення.

З діаметром тісно пов'язана інша характеристика мережі – затримка, яка рівна часу проходження повідомлення через вузли мережі до місця призначення.

Частота роботи мережі визначає, як швидко дані можуть передаватися через мережу одне за одним.

Із частотою та кількістю зв'язків мережі тісно пов'язана пропускна здатність мережі, тобто кількість інформації, яка може бути передана через мережу за одиницю часу.

Важливою характеристикою мережі є її здатність до розширення шляхом додавання додаткових вузлів. Вартість додавання вузла залежить в значній мірі від кількості компонентів і зв'язків, потрібних для додавання вузла. Для деяких мереж вартість додавання вузла фіксована. Причому, для одних мереж вартість додавання вузлів зростає лінійно або квадратично як функція розміру мережі, для інших – логарифмічно. Вартість часто є обмежуючим чинником для розширення даної мережної топології.

Надійність в значній мірі залежить від надлишковості мережі, яка визначається кількістю різних шляхів, через які повідомлення може пройти від його джерела до місця призначення. Якщо немає ніякої надлишковості, дефектний вузол може зашкодити проходженню всіх або частини повідомлень через нього, та досягненню їх місця призначення.

чення. Надлишкова мережа не тільки більш пристосована до дефектів компонентів, але також забезпечує зміну напрямів пересилання повідомлень, і таким чином полегшує вирішення питань блокування в мережі.

12.9.3. Статичні топології комунікаційних мереж багатопроцесорних систем

Статичні (фіксовані) комунікаційні мережі мають однона правлені та двонаправлені фіксовані канали зв'язку між процесорами. Може бути виділено два види статичних мереж. Це мережі з повним з'єднанням (CCN – completely connected networks) та мережі з обмеженим з'єднанням (LCN – limited connection networks).

У мережі з повним з'єднанням (повнозв'язних мережах) кожен вузол з'єднаний зі всіма іншими вузлами у мережі. Мережа з повним з'єднанням гарантує швидку доставку повідомлення від будь-якого початкового вузла до будь-якого вузла призначення, оскільки йому доведеться перетнути лише один канал зв'язку. Так як кожен вузол з'єднаний з кожним іншим вузлом в мережі, організація обміну повідомленнями між вузлами стає простим завданням. Мережі з повним з'єднанням є, проте, дорогими, оскільки вимагають великої кількості каналів зв'язку для їх побудови. Ця незручність стає очевидною при великих кількостях вузлів N . Слід зазначити, що число каналів зв'язку в мережі з повним з'єднанням рівне $N(N-1)/2$, тобто асимптотична складність цієї мережі, виражена кількістю каналів зв'язку, є $O(N^2)$. Часова затримка мережі з повним з'єднанням, виражена кількістю пересічених зв'язків, є постійною і рівною 1. Мережа з повним з'єднанням, приведена в якості прикладу на рис. 12.25, має $N = 6$ вузлів. Як результат, для їх з'єднання потрібно 15 каналів зв'язку.

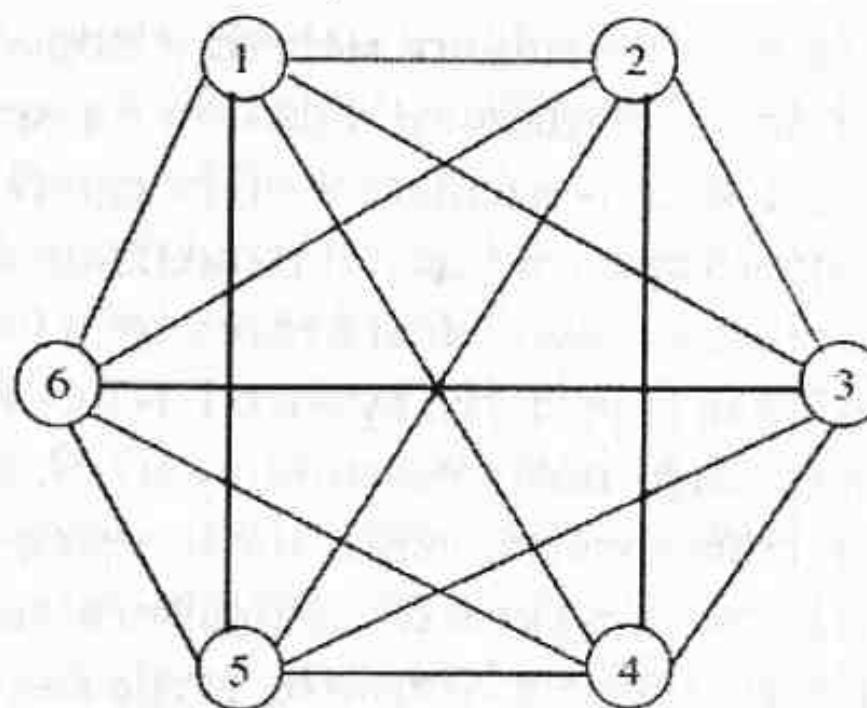


Рис. 12.25. Приклад мережі з повним з'єднанням

Мережі з обмеженим з'єднанням не забезпечують безпосереднього зв'язку кожного вузла з кожним іншим вузлом мережі. Натомість, комунікації між деякими вузлами мають бути здійснені через інші вузли мережі. Довжина шляху між вузлами, вимірювана кількістю каналів зв'язку, які доведеться перетнути, є довшою, порівняно з випадком мереж з повним з'єднанням. Крім того, при застосуванні мереж з обмеженим з'єднанням виникає ще дві інших проблеми. Це потреба пошуку ефективної топології комунікаційної мережі і потреба в механізмі маршрутизації повідомлень по мережі, поки вони не досягають своїх місць призначення.

Нижче ці дві проблеми обговорюються детальніше.

З розвитком архітектури комп'ютерів створено цілий ряд топологій комунікаційних мереж з обмеженим з'єднанням. Ці топології включають:

- одновимірні (лінійні) топології;
- двовимірні топології (кільце, зірка, дерево, решітка);
- тривимірні топології (повнозв'язна топологія, кубічна топологія, гіперкубічні топології).

Прості приклади цих топологій комунікаційних мереж показані на рис. 12.26.

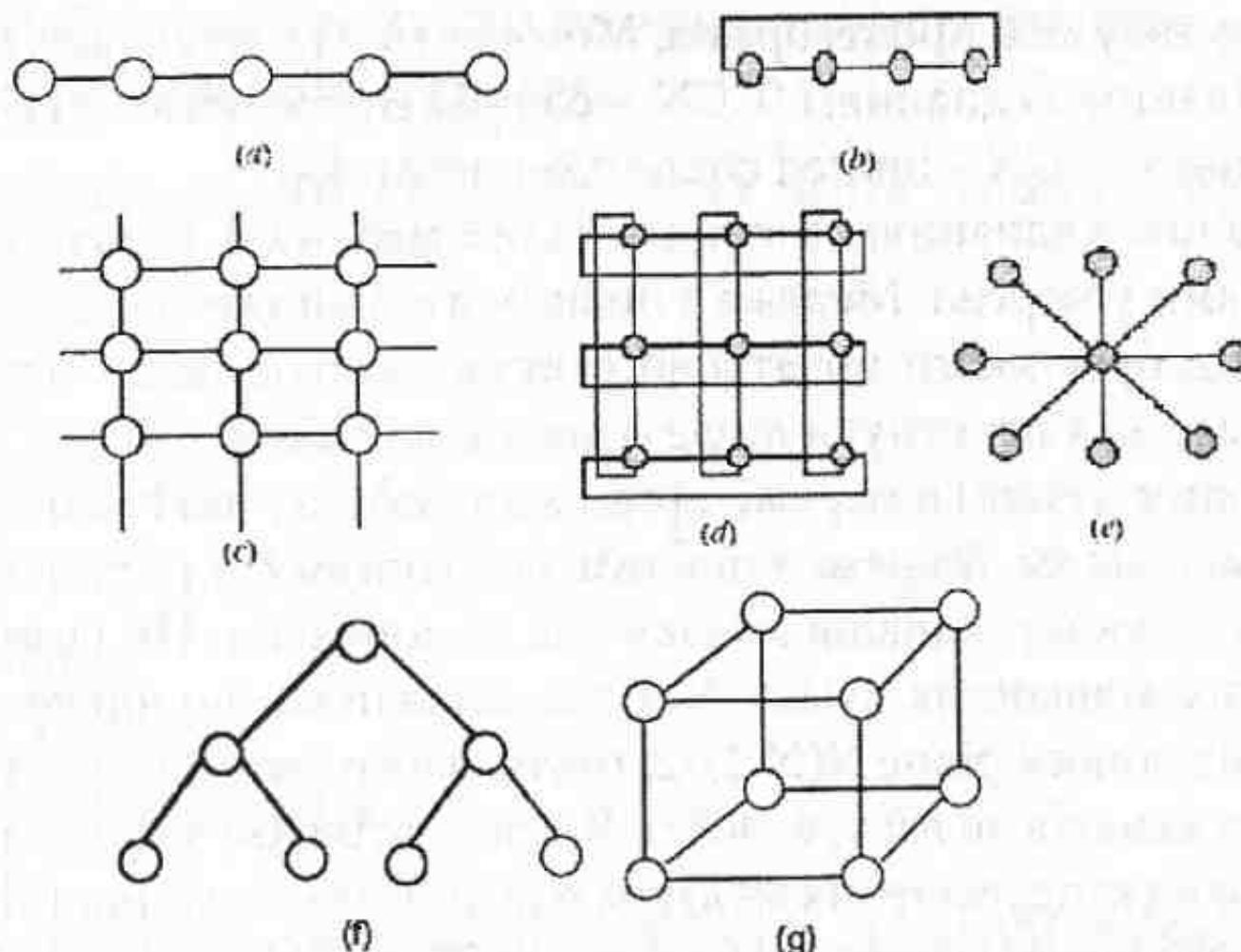


Рис. 12.26. Приклади мереж з обмеженим з'єднанням

У лінійній топології вузли комунікаційної мережі утворюють одновимірний масив (рис. 12.26 а), в якому кожен вузол з'єднаний з своїми двома безпосередніми сусідніми вузлами. Два вузли в кінцях масиву з'єднані з своїм єдиним безпосереднім сусідом. Якщо вузлу i потрібно зв'язатися з вузлом j , де $j > i$, то повідомленню від вузла i доводиться перетнути вузли $i+1, i+2, \dots, j-i$. Так само, коли вузлу i потрібно зв'язатися з вузлом j , де $i > j$, то повідомлення від вузла i має перетнути вузли $i-1, i-2, \dots, i-j$. У найгіршому випадку, коли вузлу 1 доводиться відправити повідомлення вузлу N , повідомленню доведеться перетнути в сумі $N-1$ вузлів, щоб досягти свого місця призначення. Тому, хоча лінійні масиви є простими і мають прості механізми маршрутизації, вони є повільними. Це особливо відчувається, коли число вузлів N велике. Мережна асимптотична складність лінійного масиву є $O(N)$ і його часова асимптотична складність є $O(N)$.

Якщо обидва вузли в кінцях лінійної мережі з'єднати, отримається мережа з структурою кільця (рис. 12.26 б). Вона може бути однонаправленою та двонаправленою, залежно від кількості каналів зв'язку та напрямку передачі в них даних. Прикладом систем, в яких використано топологію кільця, є мережа Token Ring та система SCI.

Решітчаста (сотова) топологія комунікаційної мережі (рис. 12.26 с) в загальному є n -вимірною матрицею, яка має $K_0 \times K_1 \times \dots \times K_{n-1}$ вузлів, де n – число вимірів мережі, і K_i є основа виміру i . Існує велика кількість сотових мереж, які відрізняються з'єднаннями вузлів. Наприклад, на рис. 12.26 d показана сотово-кільцева мережа, в якій крайні вузли замкнуті в кільце. Рис. 12.27 показує приклад сотової мережі розміром $3 \times 3 \times 2$. Вузол, позиція якого є (i, j, k) , з'єднаний із своїми сусідами з позицією $i+1, j+1, i+k+1$. Потрібно від-

значити, що для сотової мережі з N вузлами найдовша відстань між двома довільними вузлами пропорційна $N^{1/2}$, тобто її часова асимптотична складність є $O(N^{1/2})$.

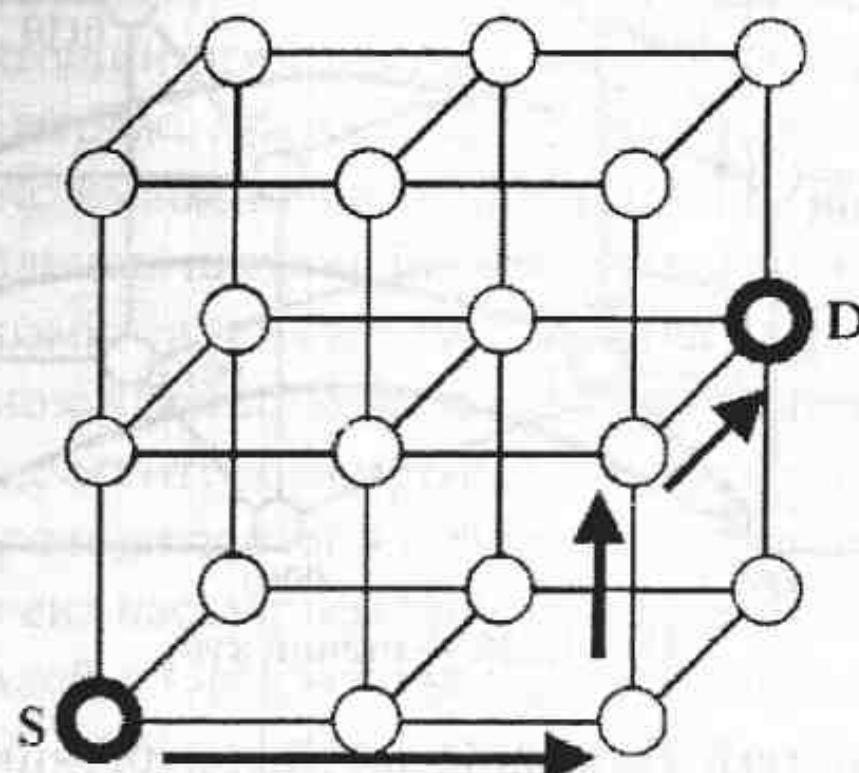


Рис. 12.27. Приклад решітчастої топології

Багатопроцесорні комп’ютерні системи з сотовою топологією комунікаційної мережі є ефективними для виконання наукових обчислень. Іншою перевагою сотових мереж є те, що вони масштабуються. Велика матриця може бути отримана з малої без зміни порядку вузла (кількості входів). Тому багато комп’ютерних систем з розподіленою пам’яттю базуються на таких мережах, наприклад система MPP фірми Goodyear Aerospace, система Paragon фірми Intel, система J-Machine Масачусетського технологічного інституту.

Зіркоподібна топологія (рис. 12.26 e) передбачає об’єднання багатьох вузлів через один центральний вузол і характеризується простотою організації керування.

В деревовидній мережі, в якій двійкове дерево, показане на рис. 12.26 f, є частковим випадком, якщо вузлу на рівні i (коренева вершина знаходиться на рівні 0) потрібно зв’язатися з вузлом на рівні j , де $i > j$ і вузол призначення належить піддереву того ж кореня, то йому доведеться направити повідомлення вгору через прохідні вузли рівнів $i-1, i-2, \dots, j+1$, поки воно не досягне вузла призначення. Якщо вузлу на рівні i потрібно зв’язатися з іншим вузлом на тому ж рівні i (або з вузлом на рівні $j \neq i$, де вузол призначення належить піддереву іншого кореня), йому доведеться направити повідомлення вгору дерева, поки повідомлення не досягає вершини на рівні 0. Після цього повідомлення доведеться направити вниз від вершини, поки воно не досягне місця призначення. Слід зазначити, що кількість вузлів в мережі двійкового дерева, що має k рівнів, рівна $N(k) = 2^k - 1$.

Максимальна глибина мережі двійкового дерева рівна $\log_2 N$, де N – число вузлів в мережі. Тому, апаратна асимптотична складність цієї мережі є $O(2^k)$ і її часова асимптотична складність є $O(\log_2 N)$.

Досить популярними є кубічні (рис. 12.26 g) та гіперкубічні топології, які дозволяють зменшити часову складність статичних топологій мереж.

Гіперкубічні мережі мають структуру n -мірного куба. n -мірний куб (гіперкуб порядку n) визначають як неорієтований граф, що має 2^n міченіх вершин від 0 до 2^{n-1} , в якому є дуга між заданою парою вершин лише в тому випадку, якщо двійкове представлення їх адрес відрізняється лише одним бітом. Як приклад, на рис. 12.28 показаний 4-мірний куб.

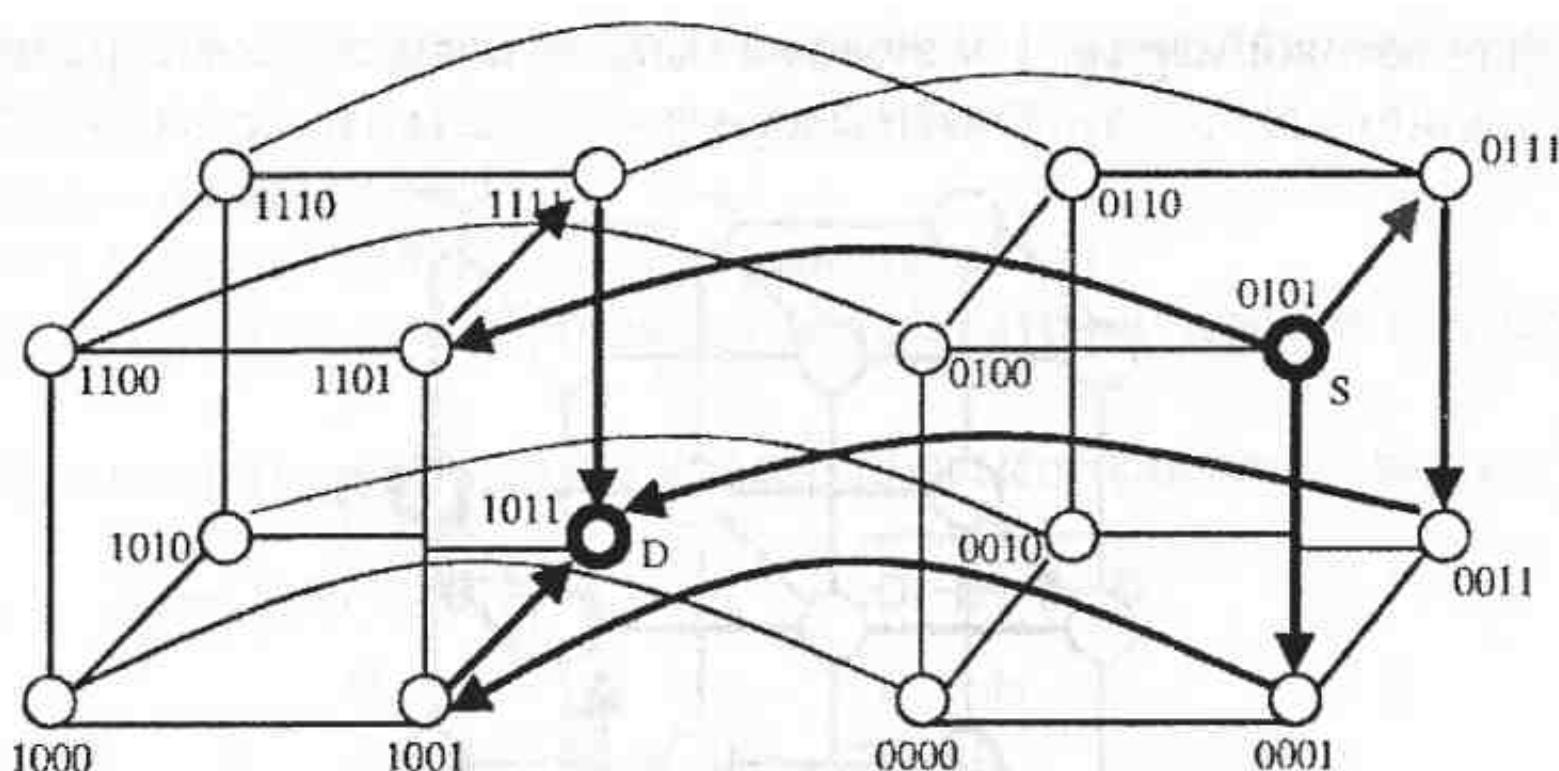


Рис. 12.28. 4-мірний куб

У багатопроцесорній системі, базованій на гіперкубічній мережі, процесори розміщуються у вершинах графа. Дуги графа представляють канали зв'язку між процесорами. Як видно з рисунку, кожен процесор в 4-мірному кубі з'єднаний з чотирма іншими процесорами. У n -мірному кубі кожний процесор має канали зв'язку з n іншими процесорами. Оскільки в гіперкубі є дуга між заданою парою вузлів лише якщо двійкове представлення їх адрес відрізняється одним бітом, ця властивість забезпечує простий механізм маршрутизації повідомлень. Маршрут повідомлення, що відправляється з вузла i та призначено для вузла j , може бути знайдений шляхом виконання над двійковим представленням адрес i та j операції виключного АБО (XOR). Якщо результат виконання операції XOR приводить до 1 в заданій бітовій позиції, то повідомлення має бути посланим по каналу зв'язку, який охоплює відповідний вимір. Наприклад, якщо повідомлення послане від початкового (S) вузла 0101 до вузла призначення (D) 1011, то результатом операції XOR буде 1110. Це означатиме, що повідомлення буде послано лише уздовж вимірів 2, 3, і 4 (рахуючи справа наліво) для того, щоб прибути до місця призначення. Порядок, в якому повідомлення перетинає три виміри, не має важливого значення. Як тільки повідомлення перетинає три виміри в будь-якому порядку, воно досягне місця призначення. Три можливих непересічних маршрути, які може пройти повідомлення у наведеному прикладі, показані виділеними лініями на рис. 12.28. Непересічні маршрути не мають будь-яких спільних каналів зв'язку.

У n -мірному кубі кожен вузол має порядок n . Порядок вузла визначений як число каналів зв'язку, що входять в вузол. Верхня межа кількості непересічних каналів в n -мірному кубі рівна n . Гіперкуб представляється як логарифмічна архітектура. Це тому, що максимальне число каналів зв'язку, які повідомленню доведеться перетнути, щоб досягти місця призначення, в n -мірному кубі, що містить $N=2^n$ вузлів, рівне $\log_2 N$.

Одна з бажаних особливостей гіперкубічних мереж – рекурсивна природа їх побудови. n -мірний куб може бути побудований на основі двох підкубів, кожен з яких має порядок $n-1$, шляхом з'єднання вузлів подібних адрес в обох підкубах. Так 4-мірний куб, показаний на рис. 12.28, побудований на основі двох підкубів, порядок кожного з яких рівний трьом. Потрібно відзначити, що побудова 4-мірного куба на основі двох 3-мірних кубів вимагає збільшення порядку кожного вузла.

Система iPSC фірми Intel є прикладом базованої на гіперкубічній комунікаційній мережі комерційної багатопроцесорної комп'ютерної системи.

12.9.4. Шинні динамічні комунікаційні мережі багатопроцесорних систем

Як ми вже вказували, динамічні комунікаційні мережі багатопроцесорних систем можуть бути одно- та багатошинними. На рис. 12.29 наведено приклад використання одношинної комунікаційної мережі, яка є найпростішим варіантом з можливих з'єднань. В загальному вигляді така система включає N процесорів P_i , кожен з яких має свою локальну пам'ять, які з'єднані спільною шиною. Всі процесори через шину взаємодіють з спільною пам'яттю. Типовий розмір таких систем – від одного до 50 процесорів. Він визначається потрібною пропускною здатністю шини, яка приходиться на один процесор. Тому описана архітектура, при простоті розширення, обмежена пропускною здатністю шини, по якій одночасно може проводити обмін інформацією лише один процесор. Складність цієї мережі може бути оцінена наступним чином: апаратна асимптотична складність, виражена кількістю магістралей, є $O(1)$, часова асимптотична складність, яка вимірюється кількістю затримок, є $O(N)$, де N – кількість процесорів.



Рис. 12.29. Одношинна багатопроцесорна система

В табл. 12.1 наведено характеристики деяких комерційних одношинних багатопроцесорних комп'ютерних систем.

Таблиця 12.1

Назва системи	Максимальна кількість процесорів	Тип процесора	Частота	Ємність пам'яті	Пропускна здатність шини
HP 9000 K640	4	PA-8000	180 MHz	4,096 MB	960 MB/s
IBM RS/6000 R40	8	PowerPC 604	112 MHz	2,048 MB	1,800 MB/s
Sun Enterprise 6000	30	UltraSPARC I	167 MHz	30,720 MB	2,600 MB/s

Використання багатьох шин для з'єднання процесорів та блоків пам'яті є природнім розширенням одношинних систем. В цьому випадку можливі кілька варіантів схем з'єднань. Серед них можна відзначити наступні:

- багатошинна комунікаційна мережа з повним з'єднанням блоків пам'яті (multiple bus with full bus-memory connection),
- багатошинна комунікаційна мережа з одним з'єднанням блоків пам'яті (multiple bus with single bus memory connection),
- багатошинна комунікаційна мережа з частковим з'єднанням блоків пам'яті (multiple bus with partial bus-memory connection),
- багатошинна комунікаційна мережа з базованим на класах з'єднанням блоків пам'яті (multiple bus with class-based memory connection).

В першому варіанті з'єднання, наведеному на рис. 12.30, всі блоки пам'яті під'єднані до всіх шин. Тут, як і на наступних рисунках, прийнято, що кількість процесорів $N = 6$, кількість модулів пам'яті $M = 4$, кількість шин $B = 4$.

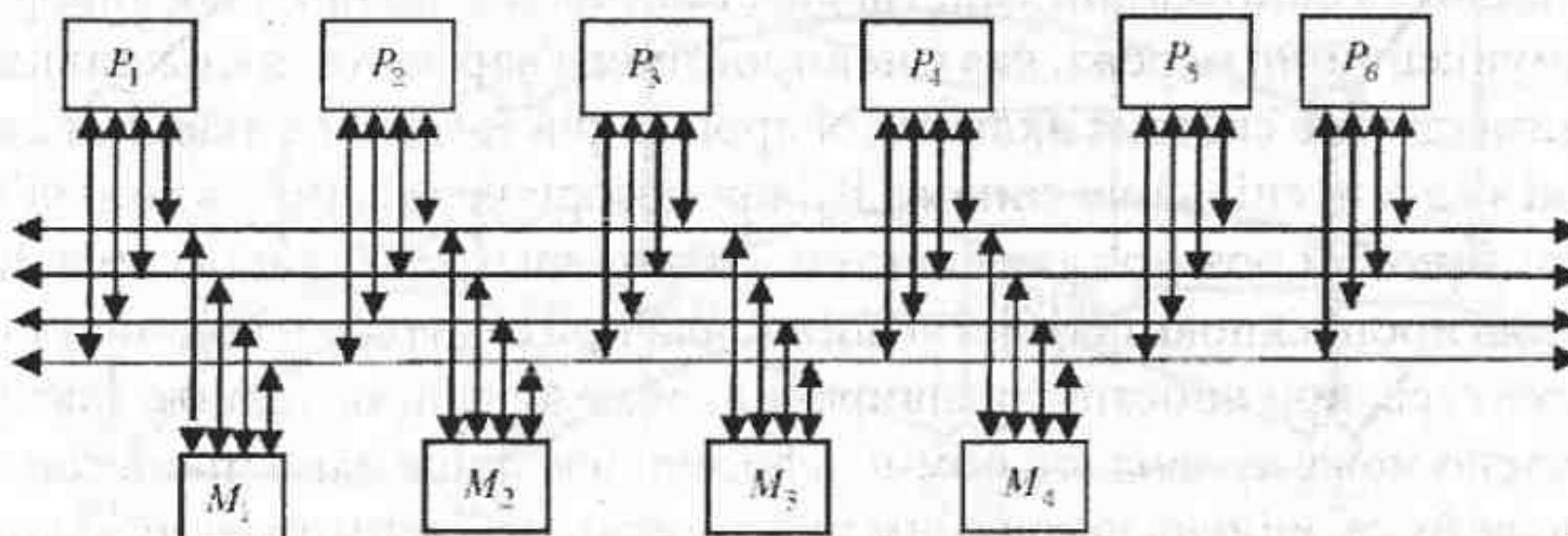


Рис. 12.30. Багатошинна комунікаційна мережа з повним з'єднанням блоків пам'яті

Цей варіант з'єднання є досить складним та дорогим, оскільки вимагає великої кількості шин, реалізація яких вимагає значних апаратних ресурсів.

В другому варіанті (рис. 12.31) кожний модуль пам'яті під'єднаний до деякої однієї шини.

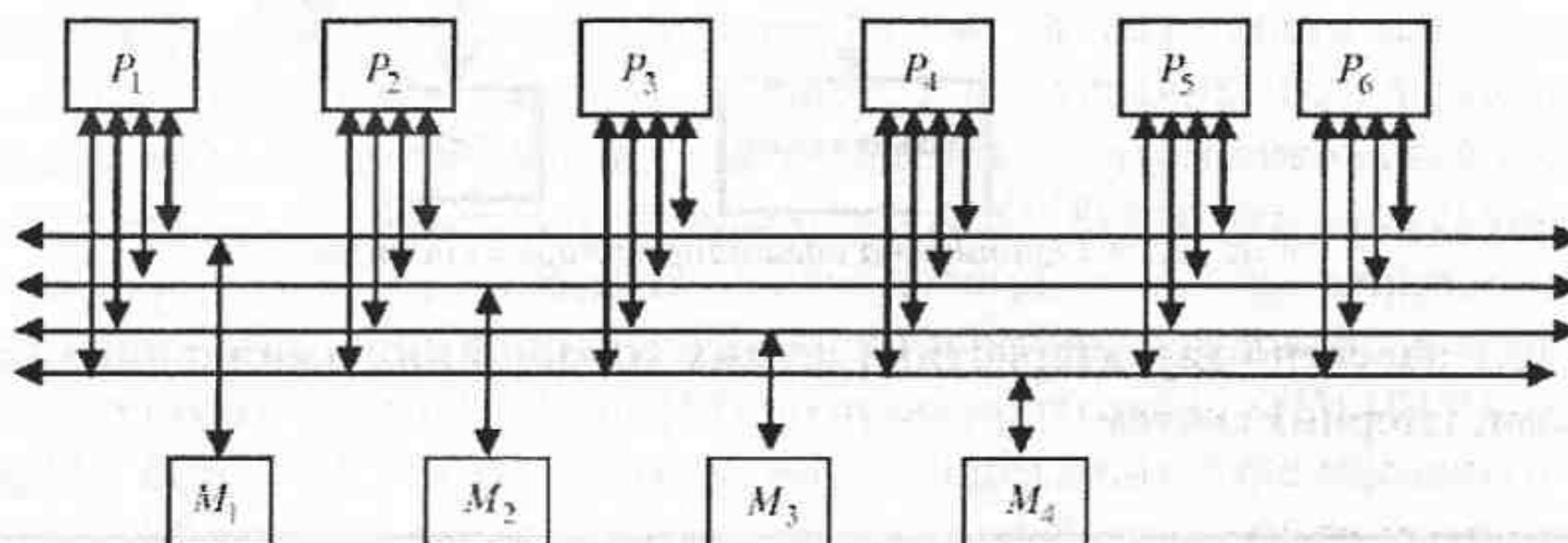


Рис. 12.31. Багатошинна комунікаційна мережа з одним з'єднанням блоків пам'яті

Тут структура комунікаційної мережі спрощується, однак можливі конфлікти доступу до блоків пам'яті.

В третьому варіанті (рис. 12.32) кожен модуль пам'яті під'єднаний не до всіх, як в першому варіанті, а до деякої групи шин (в даному випадку до двох шин).

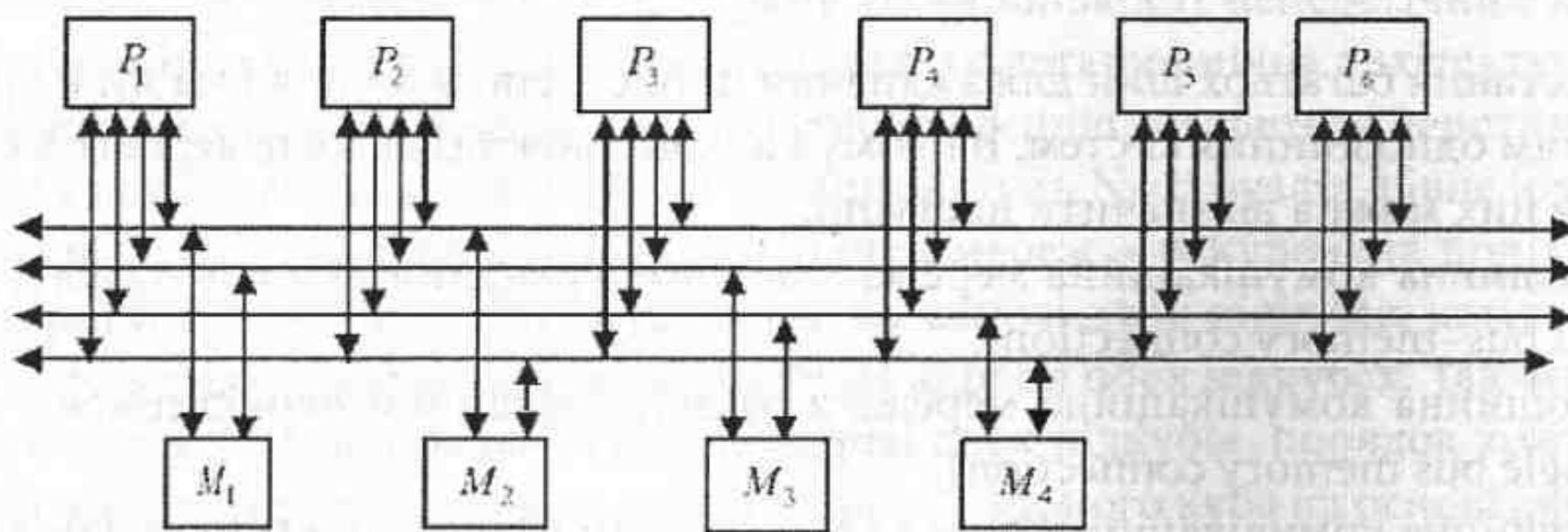


Рис. 12.32. Багатошинна комунікаційна мережа з частковим з'єднанням блоків пам'яті

Це дозволяє зменшити кількість можливих конфліктів доступу до блоків пам'яті.

В четвертому варіанті (рис. 12.33) модулі пам'яті згруповані в класи, які під'єднані до свого набору шин.

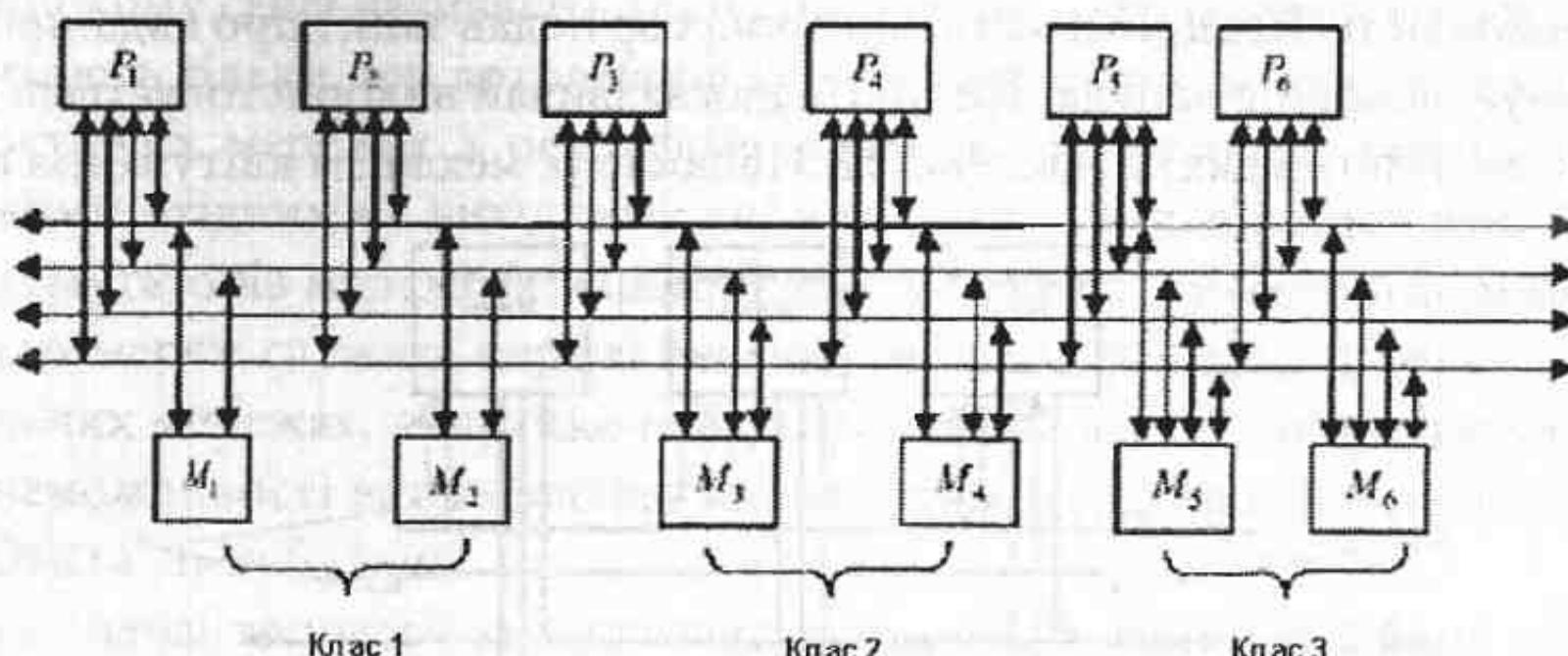


Рис. 12.33. Багатошинна комунікаційна мережа з базованим на класах з'єднанням блоків пам'яті

Тут клас – це довільно вибрана множина модулів пам'яті. Такий підхід також дозволяє зменшити кількість конфліктів шляхом врахування особливостей виконуваних задач (інтенсивності обміну процесорів з пам'яттю). Можна характеризувати цей тип мережі використовуючи кількість необхідних зв'язків і навантаження на кожнушину, як показано в табл. 12.2, де наведені характеристики розглянутих багатошинних з'єднань. У цій таблиці k представляє число класів; g представляє число шин на групу, і M_j представляє число модулів пам'яті в класі j .

Таблиця 12.2

Тип з'єднання	Кількість з'єднань	Навантаження на шину i
З повним з'єднанням	$B(N + M)$	$N + M$
З одним з'єднанням	$BN + M$	$N + M_j$
З частковим з'єднанням	$B(N + M/g)$	$N + M/g$
З базованим на класах з'єднанням	$BN + \sum_{j=1}^k M_j(j + B - k)$	$N + \sum_{j=\max(i+k-B,1)}^k M_j, 1 \leq i \leq B$

В цілому багатошинна багатопроцесорна організація має кілька бажаних особливостей, таких як висока надійність і легкість нарощування. Коли кількість шин є меншою за кількість модулів пам'яті (або кількості процесорів), змагання за владіння шиною посилюється.

Важливим питанням в багатошинній багатопроцесорній організації є шинна синхронізація. Інформація може бути передана по шині синхронно або асинхронно. Час для будь-якої передачі по синхронній шині відомий наперед. При прийманні та відправленні інформації через шину потрібно брати до уваги час передачі. Організація асинхронної шини, з другого боку, залежить від наявності даних і готовності пристрою ініціювати шинні операції.

В одношинній багатопроцесорній системі потрібен шинний арбітраж для вирішення шинного конфлікту, який має місце, коли більш ніж один процесор конкурує за владіння шиною. В даному випадку процесори, які хочуть використовувати шину, подають запити до шинної арбітражної логіки. Арбітр вирішує, використовуючи схему пріоритету, якому процесору буде надано доступ до шини протягом певного часового інтервалу.

Процес асинхронної взаємодії одного процесора з іншим називають квітуванням. Він вимагає використання трьох керуючих сигналів: запит шини, надання шини та інформування про її зайнятість. Перший вказує, що процесор подав запит про надання шини, тоді як другий вказує, що шина надана. Третій сигнал зазвичай використовується, щоб вказати, чи зайнятою є шина в даний час. Рис. 12.34 ілюструє механізм квітування шини.

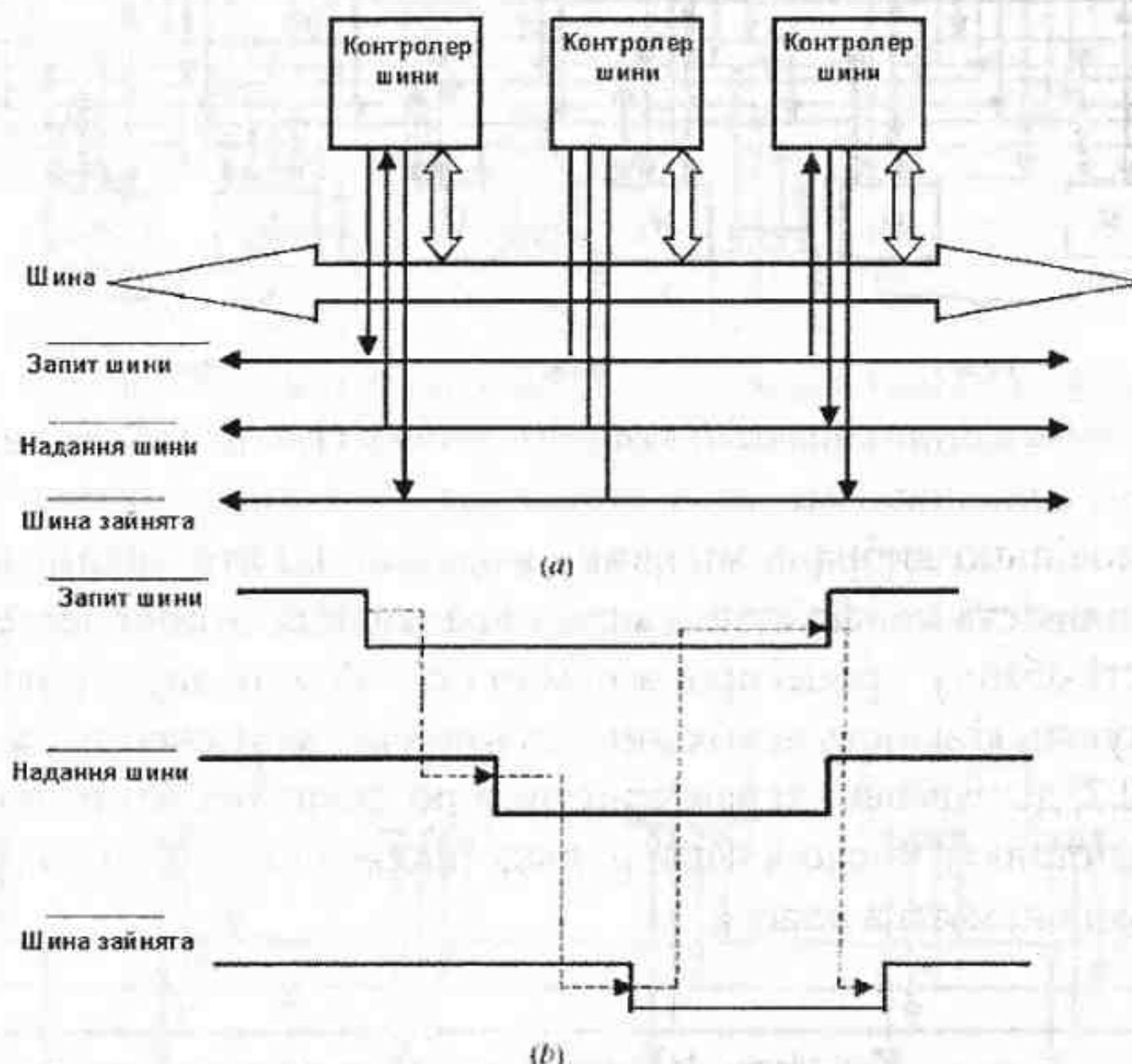


Рис. 12.34. Механізм квітування шини: а – схема, б – часова діаграма

Для прийняття рішення якому процесору надати контроль над шиною, шинна арбітражна логіка використовує будовану схему пріоритету. Використовують різні типи схем пріоритету, наприклад випадковий пріоритет, одинаковий пріоритет, пріоритет найдовше не використованого LRU.

У одинаковому пріоритеті, коли два або більше запитів зроблено, є рівний шанс у будь-якого з них. При використанні правила LRU найвищий пріоритет надано процесору, який не використав шину протягом найдовшого часу.

12.9.5. Комутуючі динамічні комунікаційні мережі багатопроцесорних систем

12.9.5.1. Типи комутуючих динамічних комунікаційних мереж

В мережах цього типу з’єднання між процесорами та модулями пам’яті здійснюється використовуючи прості комутуючі елементи.

Всі топології комутуючих комунікаційних мереж багатопроцесорних систем розділяють на три типи: неблокуючі, неблокуючі з реконфігурацією і блокуючі.

У неблокуючих мережах забезпечується з’єднання між будь-якими парами вхідних і вихідних терміналів без зміни режиму роботи комутуючих елементів мережі. В рамках цієї групи розрізняють мережі строго неблокуючі та неблокуючі в широкому сенсі.

У строго неблокуючих мережах виникнення блокувань принципово неможливе через застосовану топологію. До таких належать матрична мережа та мережа Клоса. Неблокуючими в широкому сенсі називають топології, в яких конфлікти при будь-яких з'єднаннях не виникають тільки при дотриманні певного алгоритму маршрутизації.

У неблокуючих мережах з реконфігурацією також можлива реалізація з'єднання між довільними вхідними і вихідними терміналами, але для цього необхідно змінити настройку комутаторів мережі та маршрут зв'язку між сполученими терміналами. Прикладами таких мереж служать мережі Бенеша, Бетчера, "Мемфіс" й інші.

У блокуючих мережах, якщо яке-небудь з'єднання вже встановлене, це може стати причиною неможливості встановлення інших з'єднань. До блокуючих належать мережі "Баньян", "Омега", p -куб і інші.

Існує три базові топології комутуючих динамічних комунікаційних мереж багато-процесорних систем: координатні, одноярусні та багатоярусні.

12.9.5.2. Координатна мережа

Координатна мережа забезпечує одночасне з'єднання всіх входів з усіма виходами. Вона містить комутуючий елемент (КЕ) на перетині будь-яких двох ліній. На рис. 12.35 наведено приклад координатної мережі розміром 8×8 . В цьому випадку КЕ знаходяться на кожному з 64 перетинів. На рисунку показано випадок, коли забезпечується одночасне з'єднання між входами P_i та виходами M_{8-i} для i від 1 до 8. Два можливих стани КЕ показано внизу рисунка: прямо та навхрест.

В загальному для координатної мережі розміром $N \times N$ кількість КЕ рівна N^2 , тобто апаратна асимптотична складність, виражена кількістю КЕ, є $O(N^2)$, тоді як часова асимптотична складність є $O(1)$.

Потрібно зауважити, що координатна мережа є неблокуючою.

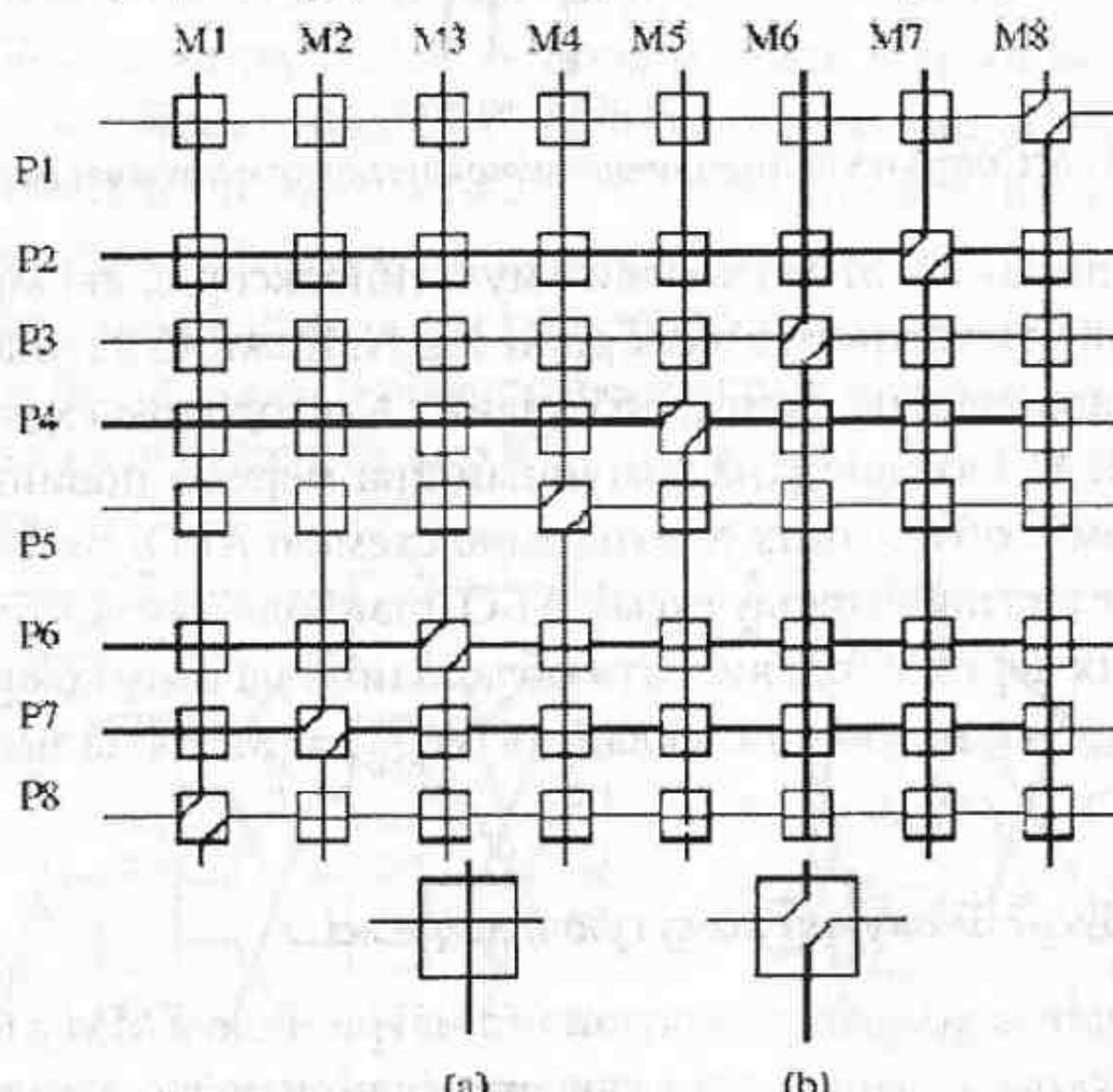


Рис. 12.35. Координатна мережа розміром 8×8 та два можливих стани КЕ:
a – прямо та b – навхрест

Топологія комутуючої комунікаційної мережі на основі матричного координатного комутатора є класичним прикладом одноярусної динамічної мережі. Головна перевага даної топології полягає в тому, що мережа є неблокуючою і забезпечує меншу затримку передачі повідомлень в порівнянні з іншими топологіями, оскільки будь-який шлях містить тільки один ключ. Проте через те, що число ключів в мережі рівне $N \times M$, використання координатного комутатора у великих мережах стає непрактичним, хоча це достатньо хороший вибір для малих мереж. Нижче буде показано, що для великих неблокуючих мереж можна запропонувати інші топології, що вимагають істотно меншої кількості ключів.

12.9.5.3. Матрична одноярусна комутуюча мережа

Матрична одноярусна комутуюча мережа є найшвидшою. Вона забезпечує реалізацію всіх типів з'єднань і є неблокуючою. Структура матричної одноярусної комутуючої мережі наведена на рис. 12.36.

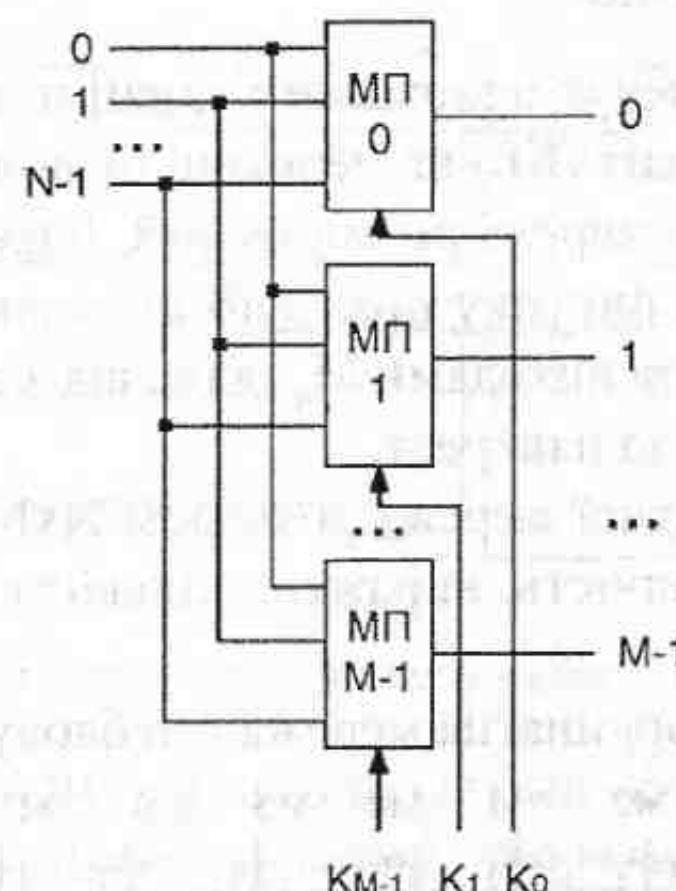


Рис. 12.36. Структура матричної одноярусної комутуючої мережі

Дана мережа складається з M N -входових мультиплексорів, які керуються M кодами K_0, K_1, \dots, K_{M-1} , розрядність кожного з яких рівна $\log_2 N$. Кожний з мультиплексорів реалізується на основі N двовходових схем І, об'єднаних N -входовою схемою АБО. Тому для комутації N входів на M виходів дана комунікаційна мережа повинна містити M груп по N двовходових схем І, об'єднаних N -входовою схемою АБО. Витрати обладнання на двовходову схему І, як і на двовходову схему АБО, рівні одному вентилю. Позначимо затримку одного вентиля через t . Тоді витрати обладнання на дану комунікаційну мережу, без врахування витрат на керування, складуть $W_{KMM} = M(2N-1)$ вентилів, а затримка $T_{KMM} = (\log N+1)t$.

12.9.5.4. Багатоярусні блокуючі комутуючі мережі

Економнішими порівняно з координатною та матричною КММ є багатоярусні комутуючі мережі, які будується на базі комутуючого елементу, що зазвичай має два входи і два виходи. Можливі комбінації з'єднання входів з виходами такого КЕ показані на рис. 12.36.

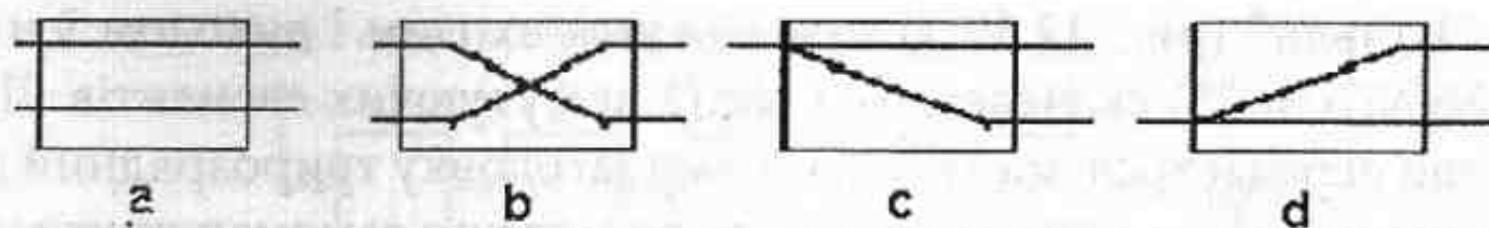


Рис. 12.36. Можливі комбінації з'єднання входів з виходами двовходового комутаційного елемента:
a – прямо, b – навхрест, c – розширення зверху, d – розширення знизу

Багатоярусна комунікаційна мережа складається з деякої множини ярусів, побудованих на двохходових КЕ, та об'єднаних між ярусними зв'язками (МЗ), як це показано на рис. 12.37. Ці зв'язки можуть відображати одну з можливих функцій маршрутизації, таку як батерфляй, куб і т. д.

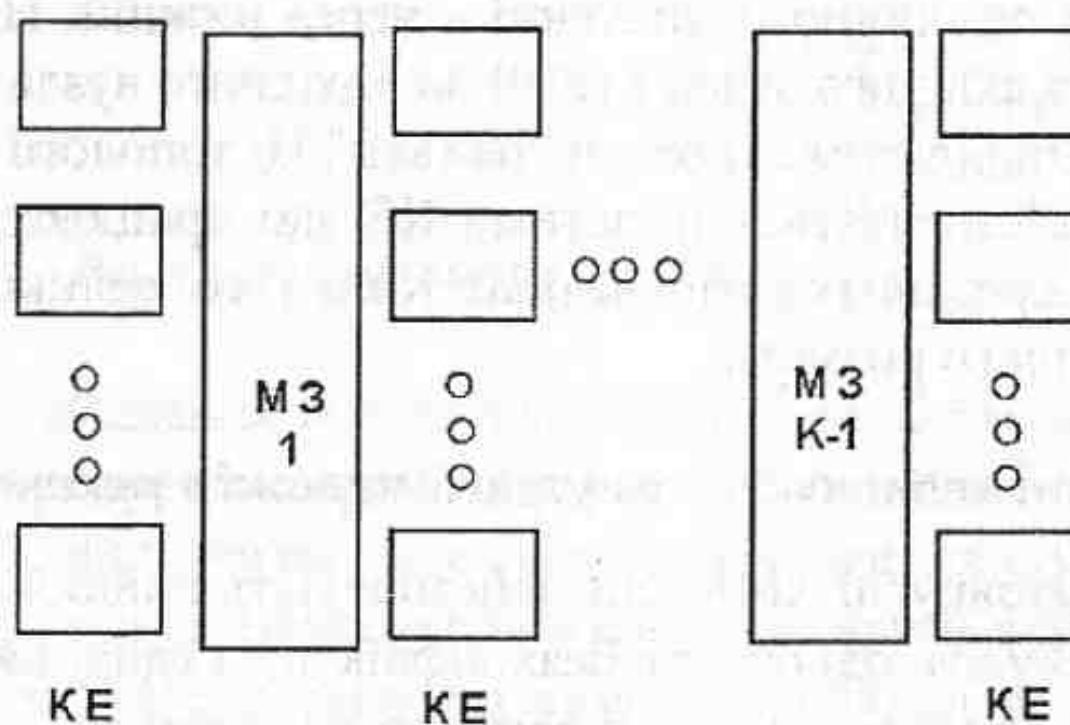


Рис. 12.37. Багатоярусна комутуюча комунікаційна мережа

Існує цілий ряд багатоярусних комутуючих комунікаційних мереж. Структура широко розповсюджених топологій мереж “Баньян” та “Омега” для $N=8$ показана відповідно на рис. 12.38 а та на рис. 12.38 б. Вона містить в кожному з $\log_2 N$ ярусів по $N/2$ КЕ і N каналів зв'язку. Якщо кожен КЕ виконує перемикання прямо і навхрест, то така комунікаційна мережа може виконати $2^{N/2\log N}$ перестановок, що істотно менше за $N!$ перестановок, можливих в неблокуючій мережі. Проте ті перестановки, які вона виконує, є найбільш використовуваними в багатопроцесорних системах. Потрібно відзначити, що в даній КММ є можливість її розділення на комунікаційні мережі меншого розміру шляхом включення на передачу прямо КЕ в тих ярусах, що стоять перед цими комунікаційними мережами.

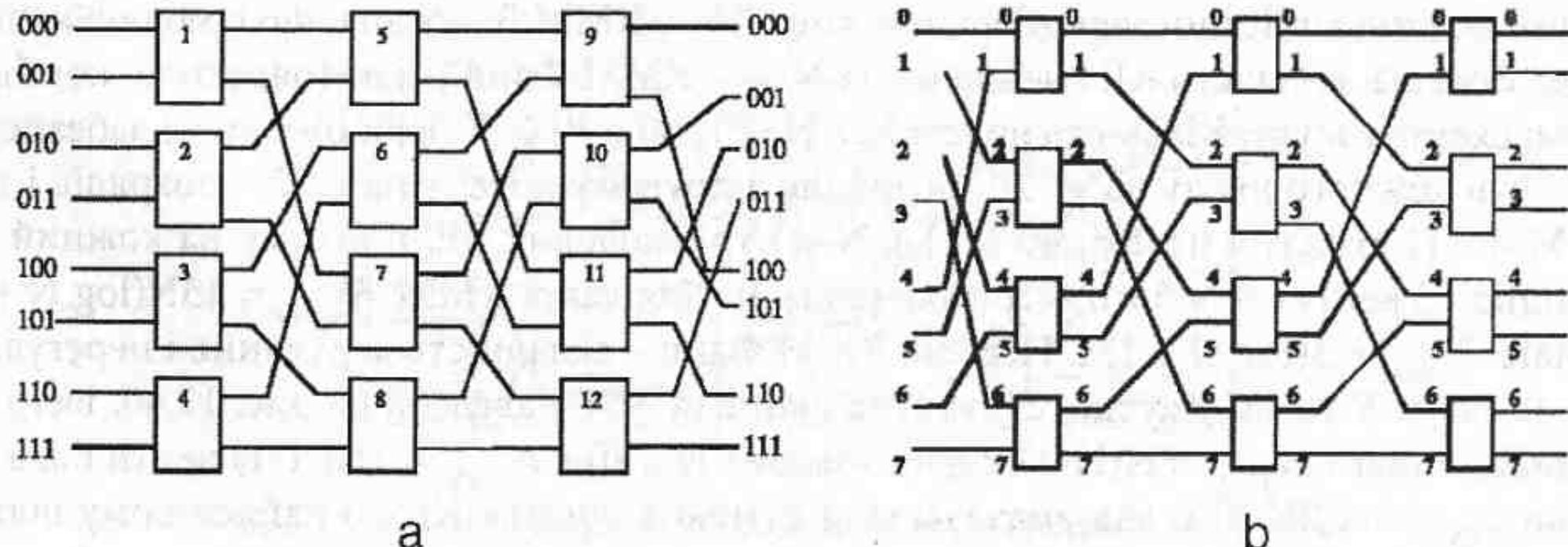


Рис. 12.38. Багатоярусна комунікаційна мережа 8x8: a – “Баньян” та b – “Омега”

У топології “Баньян” (рис. 12.38 а) між кожним входом і виходом існує лише один шлях. Мережа $N \times N$ ($N = 2^m$) складається з $Nm/2$ комутуючих елементів. Для керування мережею пакет, що передається, містить в своєму заголовку трироздільний двійковий номер вузла призначення. Дані мережа належить до мереж з самомаршрутизацією, оскільки адреса пункту призначення не тільки визначає маршрут повідомлення до потрібного вузла, але і використовується для керування проходженням повідомлення по цьому маршруту. Кожен КЕ, до якого потрапляє пакет, проглядає один біт адреси і, залежно від його значення, направляє повідомлення на вихід 1 або 2. Стан КЕ першого ярусу мережі (лівий стовпець КЕ) визначається старшим бітом адреси вузла призначення. Середнім ярусом (другий стовпець) управляє середній біт адреси, а третім ярусом (правий стовпець) – молодший біт. Якщо значення біта рівне 0, то повідомлення пропускається через верхній вихід КЕ, а при одиничному значенні – через нижній. На рисунку показаний маршрут повідомлення з вхідного вузла 2 (010) до вихідного вузла 5 (101).

Топологія “Омега” є підкласом топології “Баньян”. Ці топології досить популярні через те, що комутація забезпечується простими КЕ, що працюють з однаковою швидкістю, а повідомлення передаються паралельно. Крім того, великі мережі можуть бути побудовані з мереж меншого розміру.

12.9.5.5. Багатоярусні неблокуючі комутуючі мережі з реконфігурацією

Розглянемо далі багатоярусні КММ, що забезпечують повний набір з $N!$ перестановок. Дані КММ забезпечують одночасне безконфліктне з'єднання довільного виходу з довільним входом, і тому представляють підвищений інтерес.

Однією з найбільш відомих і вивчених КММ даного класу є КММ Бенеша, структура якої для $N=8$ наведена на рис. 12.39.

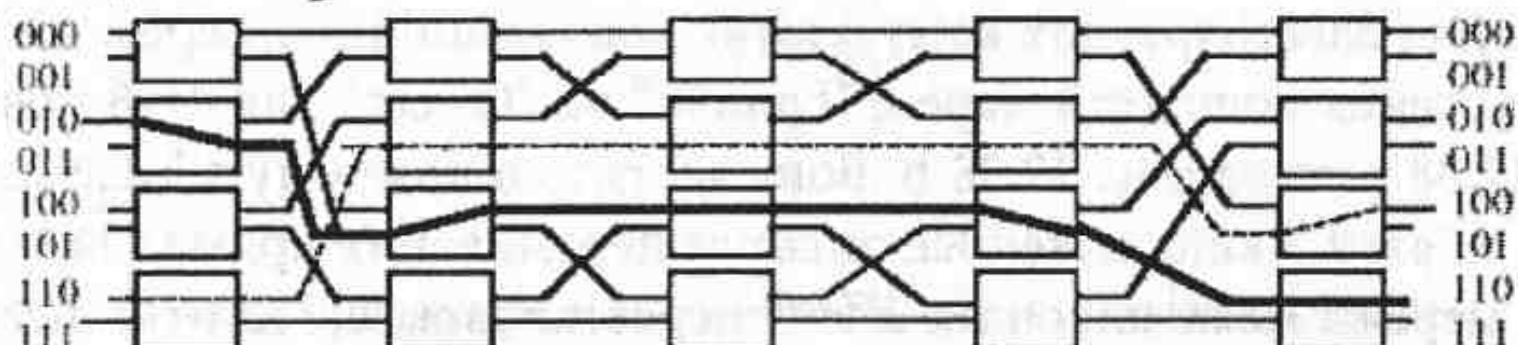


Рис. 12.39. Багатоярусна неблокуюча комутуюча мережа Бенеша з реконфігурацією

Для даної мережі $W_{\text{КММ}} = N(\log_2 N - 1/2)$ двовходових КЕ, або $W_{\text{КММ}} = 3N(\log_2 N - 1)$ вентилів, кількість ярусів $m = 2\log_2 N - 1$, а затримка $T_{\text{КММ}} = 2(2\log_2 N - 1)t$.

Існує цілий спектр інших неблокуючих комутуючих мереж із реконфігурацією з подібними до вищеописаної мережі функціями. Так, у КММ Ваксмана, що є модифікацією мережі Бенеша, кількість КЕ зменшена на $N/2 - 1$. КММ Фаня є багатоярусною комбінаційною схемою зсуву. Кількість ярусів для $N = 2^p$ рівне $p + 1$. У першому ярусі забезпечується зсув кожного входу на $+/-2^{p-1}$ розрядів, в другому ярусі – на $+/-2^{p-2}$ розрядів і т. д. На КММ з N входами необхідно $N(\log_2 N + 1)$ тривходових КЕ, причому на кожний КЕ необхідно 15 вентилів, а його затримка рівна $3t$. Для даної КММ $W_{\text{КММ}} = 15N(\log_2 N + 1)$ вентилів, $T_{\text{КММ}} = 3(\log_2 N + 1)t$. Недолік КММ Фаня – складність керування і нерегулярність зв'язків. У КММ Каутца, структура якої для $N=6$ наведена на рис. 12.40, витрати обладнання рівні $W_{\text{КММ}} = N(N-1)/2$ двовходових КЕ, або $W_{\text{КММ}} = 3N(N-1)$ вентилів, а затримка $T_{\text{КММ}} = 2(2N-3)t$, яка визначається сумою затримок КЕ по найдовшому шляху проходження даних. Перевага даної мережі – однорідність і регулярність.

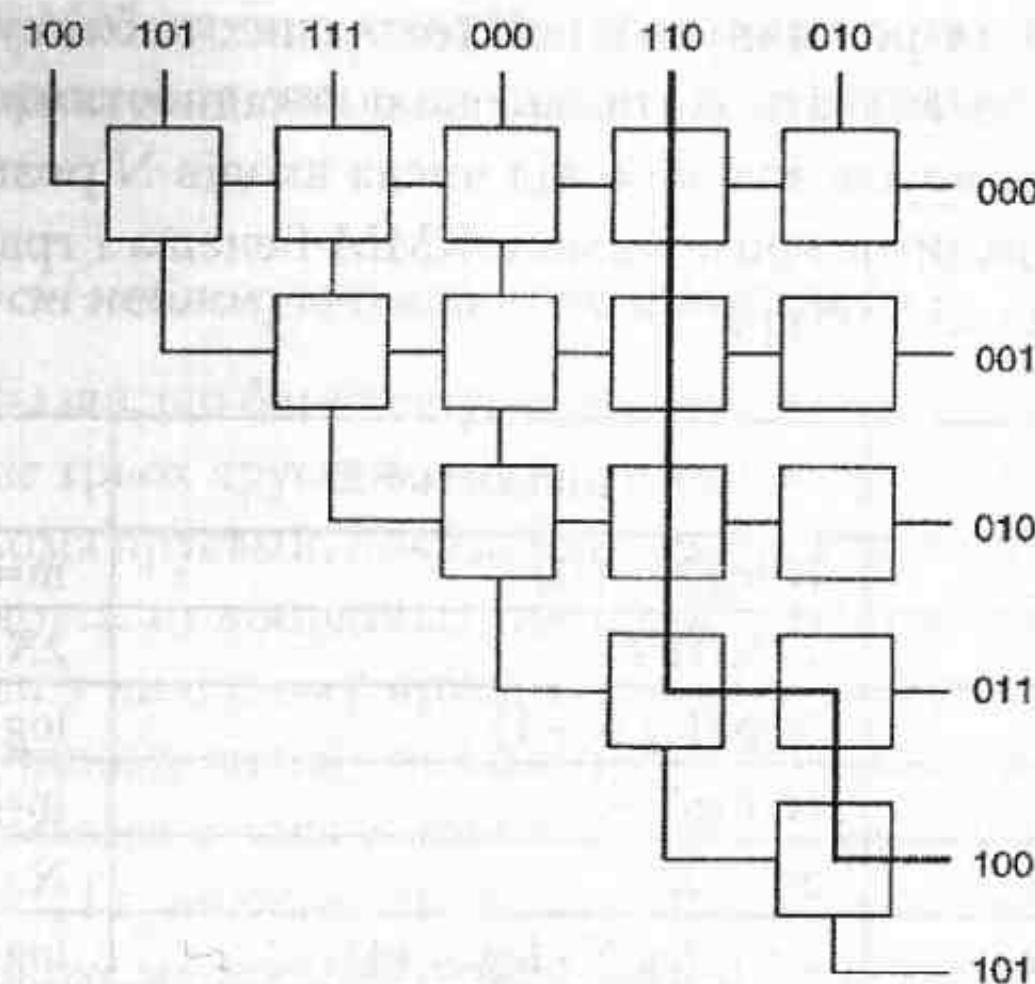


Рис. 12.40. Структура КММ Каутца для $n=6$

Одночасне довільне з'єднання всіх виходів КММ з всіма входами забезпечують КММ, створені на базі структур сортувальних мереж, що запропоновано та описано в працях автора даної книги. Потрібно відзначити, що можна синтезувати велику кількість структур багатоярусних КММ на базі сортувальних мереж з кількістю ярусів від $\log_2 N(\log_2 N+1)/2$ до $N-1$. Вибір конкретної структури залежить від вимог по швидкодії, обмежень за апаратною складністю, а також, можливо, і обмежень на топологію КММ. Розглянемо два крайні випадки даних структур. Структура КММ мінімальної швидкодії з даного класу, назовемо її крайньою зліва, аналогічна структурі відповідного графа сортування (рис. 12.41 а), має характеристики: $W_{\text{КММ}} = 6(3^{\log N} - 2^{\log N})$ вентилів, $T_{\text{КММ}} = 2(N - 1)t$. Швидша КММ, назовемо її крайньою справа, відповідна графу сортування Бетчера (рис. 12.41б), має характеристики: $W_{\text{КММ}} = 6 \cdot 2^{\log N-2}(\log^2 N - \log N + 4) - 1$ вентилів, $T_{\text{КММ}} = \log N (\log N + 1)t$.

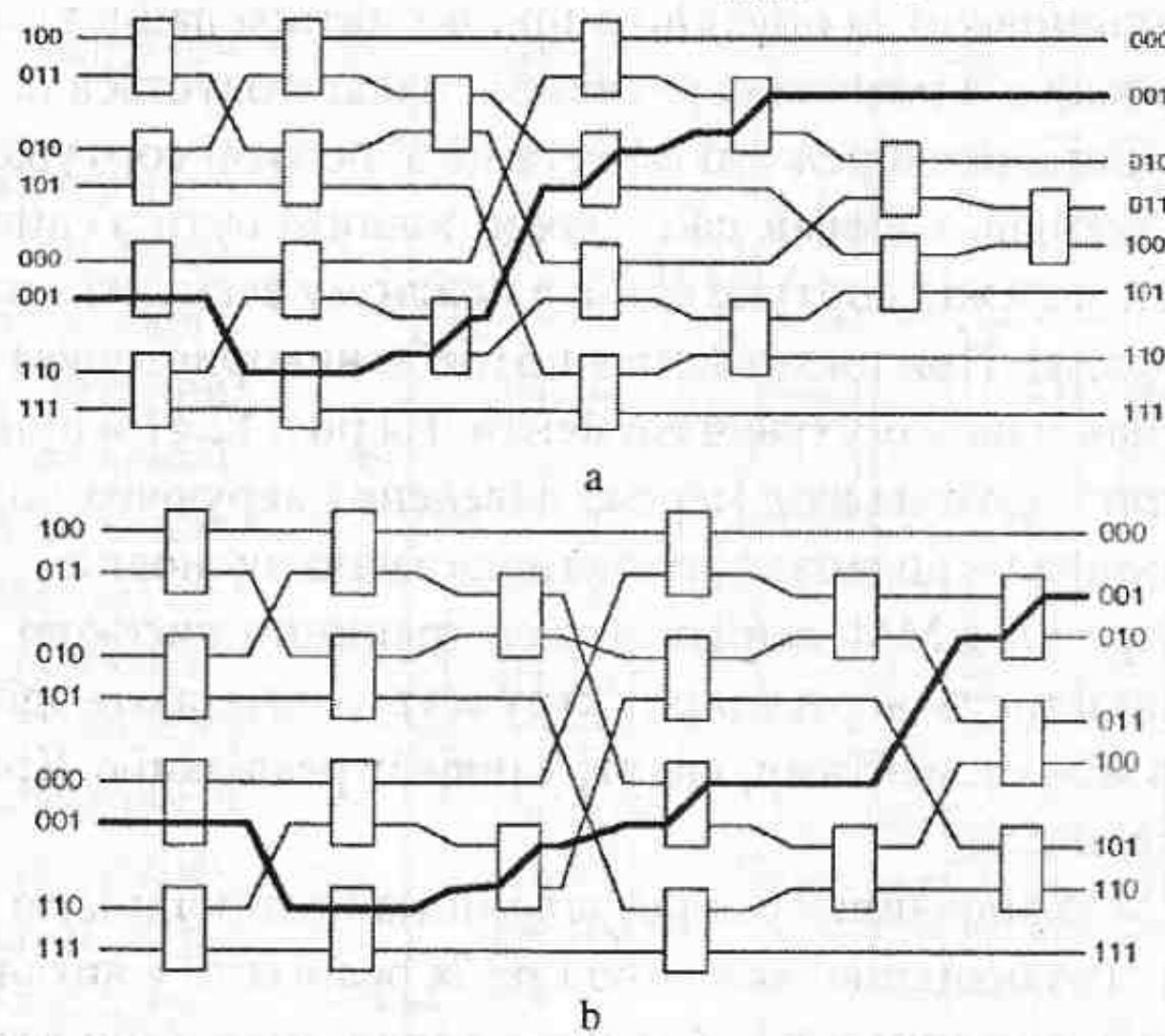


Рис. 12.41. КММ, створені на базі структур сортувальних мереж:
а – крайня зліва, б – крайня справа

Проведемо порівняння розглянутих багатоярусних неблокуючих комутуючих мереж з реконфігурацією за апаратною та часовою складностями. В табл. 12.3 наведено кількість КЕ та кількість ярусів залежно від числа входів N розглянутих КММ. Видно, що найвигіднішими за даними критеріями є КММ Бенеша і границя справа КММ на основі сортувальної мережі Бетчера.

Таблиця 12.3

Тип мережі	Кількість КЕ	Кількість ярусів
Бенеша	$N(\log N - 1/2)$	$m=2\log_2 N - 1$
Каутца	$N(N-1)/2$	$2N-3$
Фаня	$5/3N(\log_2 N + 1)$	$\log N + 1$
Ваксмана	$N(\log N - 1)$	$m=2\log_2 N - 1$
Сортувальна крайня зліва	$3^{\log N} - 2^{\log N}$	$N - 1$
Сортувальна крайня справа	$2^{\log N-2}(\log^2 N - \log N + 4)$	$\log N(\log N + 1).2$

Розглянемо принципи керування багатоярусними КММ. У матричній одноярусній КММ керуючий код для кожного з N мультиплексорів (приймемо, що кількість входів КММ рівна кількості виходів) займає $\log_2 N$ розрядів. Він дешифрується дешифратором мультиплексора і пропускає на вихід КММ інформацію з необхідного входу. У багатоярусних КММ кількість бітів керуючої інформації також рівна $N \log_2 N$, але в них $\log_2 N$ -роздрядний код рухається разом з інформаційним кодом і настроює КЕ на необхідний вид перемикань. Раніше був описаний принцип керування багатоярусною мережею “Баньян” (рис. 12.38а). Подібно в КММ “узагальнений куб” тег $T = T_m T_{m-1} \dots T_0$ формується як сума за модулем два двійкових номерів джерела $S = S_m S_{m-1} \dots S_0$ і приймача $D = D_m D_{m-1} \dots D_0$, тобто $T = S + D$. Кожний розряд тега поступає на відповідний його номеру КЕ і настроює його на режим роботи прямо, якщо $T_i = 0$, або навпаки, якщо $T_i = 1$.

Ефективно розв'язується питання керування в КММ, побудованих на базі сортувальних мереж, що запропоновано та описано в працях автора даної книги. В цьому випадку КЕ, спочатку працюючи в режимі сортування, налаштовується на необхідний режим перемикання, а потім працює в режимі комутації. У режимі сортування на входи КММ поступають адреси виходів, з якими дані входи повинні бути з'єднані. Дані адреси рухаються по елементах мережі і сортуються, а в кожному елементі запам'ятовується стан керуючого коду (розряду). При цьому немає потреби використовувати окремі шини для подачі керуючих сигналів на комутуючі елементи. На рис. 12.41 жирною лінією показано один із маршрутів при подачі на вход мережі наведених керуючих кодів.

Проведений вище аналіз дозволяє зробити наступні висновки:

- одноярусні матричні КММ відрізняються гранично високою швидкодією, проте мають недостатню надійність, нерегулярну структуру, вимагають здійснення дуже великої кількості зв'язків між елементами, що утруднює їх реалізацію. Крім того, вони мають велику апаратну складність;
- багатоярусні КММ відрізняються регулярністю і однорідністю структури, а також локальністю зв'язків, що особливо важливо при їх реалізації у вигляді НВІС. Крім того, в більшості структур багатоярусних КММ вихід з ладу одного або декількох КЕ практично не впливає на їх працездатність;

- в багатоярусних КММ на базі сортувальних мереж, які характеризуються високими параметрами апаратної та часової складності, ефективно розв'язується питання керування.

12.9.5.6. Багатоярусні неблокуючі комутуючі мережі

У 1953 році Клос показав, що багатоярусна мережа на основі координатних комутаторів, що містить не менше трьох ярусів, може мати характеристики неблокуючої мережі.

Мережа Клоса з трьома ярусами, показана на рис. 12.42, містить r_1 координатних комутаторів у входному ярусі, m координатних комутаторів в проміжному ярусі і r_2 координатних комутаторів у вихідному ярусі. В кожного комутатора входного яруса є n_1 входів і m виходів – по одному виходу на кожний координатний комутатор проміжного яруса. Комутатори проміжного яруса мають r_1 входів, за кількістю координатних комутаторів входного яруса, і r_2 виходів, що відповідає кількості перемикачів у вихідному ярусі мережі. Вихідний ярус мережі будується з координатних комутаторів з m входами і n_2 виходами. Звідси зрозуміло, що числа n_1, n_2, r_1, r_2 , і m повністю визначають мережу. Число входів мережі $N = r_1 n_1$, а виходів – $M = r_2 n_2$.

Зв'язки всередині складеного комутатора організовані за наступними правилами:

- k -й вихід i -го входного комутатора з'єднаний з i -м входом k -го проміжного комутатора;
- k -й вихід j -го вихідного комутатора з'єднаний з j -м виходом k -го проміжного комутатора.

Кожен модуль першого і третього ярусів мережі з'єднаний з кожним модулем другого її яруса.

Хоча в даній топології забезпечується шлях від будь-якого входу до будь-якого виходу, відповідь на питання, чи буде мережа неблокуючою, залежить від числа проміжних ланок. Клос довів, що подібна мережа є неблокуючою, якщо кількість координатних комутаторів в проміжному ярусі m задовільняє умові: $m = n_1 + n_2 - 1$. Якщо $n_1 = n_2$, то матричні перемикачі в проміжному ярусі є повними координатними комутаторами і

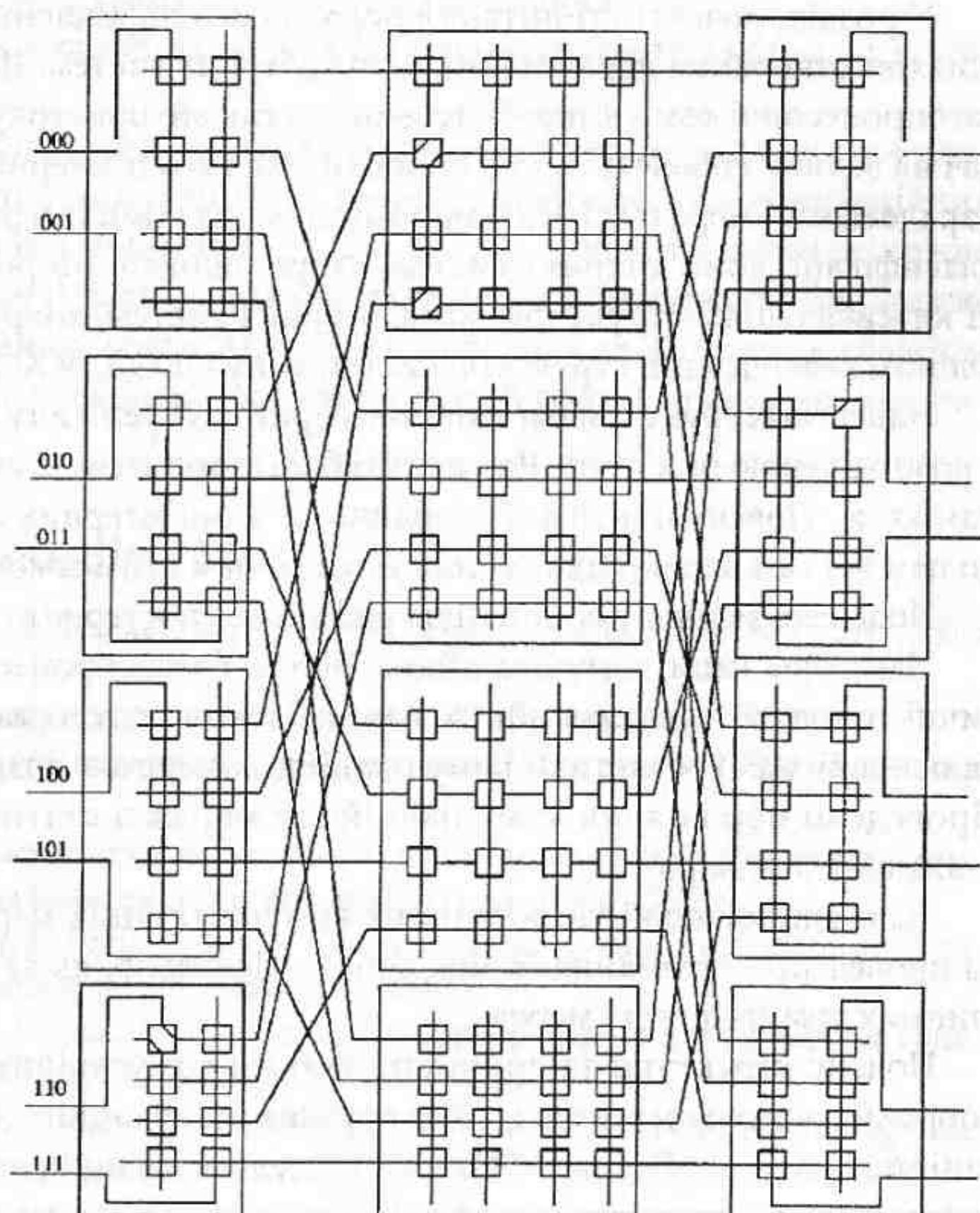


Рис. 12.42. Триярусна мережа Клоса

критерій неблокованості набуває вигляду: $m = 2n - 1$. За умови $m = n$, мережу Клоса можна віднести до неблокуючих мереж з реконфігурацією. У всіх інших випадках дана топологія стає блокуючою.

Комп'ютерні системи, в яких з'єднання реалізовані відповідно до топології Клоса, випускають багато фірм, зокрема Fujitsu, Nippon, Hitachi.

На завершення в табл. 12.4 наведено порівняння швидкодії вище розглянутих динамічних комунікаційних мереж.

Таблиця 12.4

Тип мережі	Затримка мережі	Ціна (складність)	Наявність блокування
Одношинна	$O(N)$	$O(1)$	Є
Багатошинна	$O(mN)$	$O(m)$	Є
Багатоярусна	$O(\log N)$	$O(N \log N)$	Є
Координатна	$O(1)$	$O(N^2)$	Немає

12.10. Короткий зміст розділу

У розділі розглянуті питання подальшого підвищення продуктивності комп'ютерів шляхом створення паралельних комп'ютерних систем. До таких систем належать як багатопроцесорні комп'ютерні системи, в яких використовується паралелізм розподілу задач на велику кількість вузлів обробки, так і комп'ютерні мережі, які мають іншу форму паралелізму – мережу структурно автономних комп'ютерів. Наведені основні положення класифікації комп'ютерних систем, запропоновані Шором та Фліном. Подано відповідні класифікаціям структури комп'ютерних систем, зокрема відповідні запропонованій Фліном класифікації структури систем типу ОКОД, МКОД, ОКМД та МКМД.

Наведено структури багатопроцесорних систем типу ОКМД та МКМД з спільною та з розподіленою пам'яттю. Розглянуті багатопроцесорні системи типу МКМД з спільною пам'яттю. Пояснені основні особливості комп'ютерних систем з однорідним доступом до пам'яті та з неоднорідним доступом до пам'яті, а також лише з кеш пам'яттю.

Пояснено загальну організацію роботи комп'ютерних систем з розподіленою пам'яттю.

Наведено типи комунікаційних мереж багатопроцесорних систем. Розглянуті статичні топології комунікаційних мереж багатопроцесорних систем. Подано особливості та основні характеристики комунікаційних мереж з повним та з неповним з'єднанням. Проведено порівняння комунікаційних мереж з статичним з'єднанням – 1-вимірні, 2-вимірні, 3-вимірні.

Наведено основні типи шинних комунікаційних мереж багатопроцесорних систем та проведено порівняння їх між собою. Показано, як здійснюється синхронізація шин шинних комутаційних мереж.

Подано структури та принципи роботи комутаційних комунікаційних мереж типу координатного комутатора, одноярусних комутаційних мереж, багатоярусних комутаційних мереж: "Баньян", "Омега". Наведено та оцінено структури багатоярусних неблокуючих комутуючих мереж з реконфігурацією, а також неблокуючих комутаційних мереж.

12.11. Література для подальшого читання

Питання впровадження паралелізму в архітектурі комп’ютера та побудови паралельних комп’ютерних систем розкриті в багатьох підручниках та монографіях, зокрема [1–4, 11–16, 19–22, 24, 25, 27], і в інформаційних матеріалах фірм-виробників, розміщених на їх веб-сторінках, зокрема фірм IBM, Intel, Sun Microsystems, HP і т. д.

Закон Амдала та наслідки з нього, а також аналіз обмежень на кількість процесорів в багатопроцесорних системах наведено в роботі [19].

Багатопотокова обробка, в тому числі технологія гіперпотокової обробки, описана в роботі [19], а також в описах процесора Intel Xeon.

Питання класифікації комп’ютерних систем розглянуті в роботах [1–4, 16, 19, 24] та багатьох інших.

Велика кількість літератури присвячена розгляду принципів побудови систем типу ОКМД та МКМД. В першу чергу слід звернутися до літератури [2–4, 6, 13–15, 17, 18, 20–22, 25, 33–35, 38].

В працях [5, 7, 20–22, 25, 33–35, 38] наведено типи комунікаційних мереж багатопроцесорних систем. Тут же розглянуті статичні топології комунікаційних мереж багатопроцесорних систем, особливості та основні характеристики комунікаційних мереж з повним та з неповним з’єднанням, а також із статичним з’єднанням.

Основні типи шинних комунікаційних мереж багатопроцесорних систем та їх порівняння наведено в [10, 20, 37].

Структури та принципи роботи комутаційних комунікаційних мереж типу координатного комутатора, одноярусних комутаційних мереж, багатоярусних комутаційних мереж: “Баньян”, “Омега” подано в роботах [8, 9, 20, 29, 30]. В цих же роботах можна знайти структури багатоярусних неблокуючих комутуючих мереж з реконфігурацією, а також неблокуючих комутаційних мереж. Питанням побудови та керування неблокуючими комутуючими мережами з реконфігурацією на основі сортувальних мереж присвячені роботи [39–41].

12.12. Література до розділу 12

1. Воеводин В. В., Воеводин В. В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002 – 608 с.
2. Головки Б. А. Параллельные вычислительные системы. – М.: Наука, 1980. – 520 с.
3. Мультипроцессорные системы и параллельные вычисления/ Под ред. Ф. Г. Энслоу. – Мир, 1976. – 384 с.
4. Тербер К. Дж. Архитектура высокопроизводительных вычислительных систем / Пер. с англ. – М.: Наука. Главная редакция физико-математической литературы, 1985. – 272 с.
5. Abraham, S. and Padmanabhan, K. Performance of the direct binary n-cube network for multiprocessors. IEEE Transactions on Computers, 38 (7), 1000–1011 (1989).
6. Agrawal, P., Janakiram, V. and Pathak, G. Evaluating the performance of multicomputer configurations. IEEE Transaction on Computers, 19 (5), 23–27 (1986).
7. Al-Tawil, K., Abd-El-Barr, M. and Ashraf, F. A survey and comparison of wormhole routing techniques in mesh networks. IEEE Network, March/April 1997, 38–45 (1997).
8. Bhuyan, L. N. (ed.) Interconnection networks for parallel and distributed processing. Computer (Special issue), 20 (6), 9–75 (1987).

9. Bhuyan, L. N., Yang, Q. and Agrawal, D. P. Performance of multiprocessor interconnection networks. *Computer*, 22 (2), 25–37 (1989).
10. Chen, W.-T. and Sheu, J.-P. Performance analysis of multiple bus interconnection networks with hierarchical requesting model. *IEEE Transactions on Computers*, 40 (7), 834–842 (1991).
11. Dasgupta, S. *Computer Architecture: A Modern Synthesis*, vol. 2; Advanced Topics, John Wiley, 1989.
12. Decegama, A. *The Technology of Parallel Processing: Parallel Processing Architectures and VLSI Hardware*, Vol. 1, Prentice-Hall, 1989.
13. Dongarra, J. *Experimental Parallel Computing Architectures*, North-Holland, 1987.
14. Duncan, R. A survey of parallel computer architectures. *Computer*, 23 (2), 5–16 (1990).
15. El-Rewini, H. and Lewis, T. G. *Distributed and Parallel Computing*, Manning & Prentice Hall, 1998.
16. Flynn. *Computer Architecture: Pipelined and Parallel Processor Design*, Jones and Bartlett, 1995.
17. Goodman, J. R. Using cache memory to reduce processor-memory traffic. *Proceedings 10th Annual Symposium on Computer Architecture*, June 1983, pp. 124–131.
18. Goyal, A. and Agerwala, T. Performance analysis of future shared storage systems. *IBM Journal of Research and Development*, 28 (1), 95–107 (1984).
19. Hennessy, J. and Patterson, D. *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 1990.
20. Hesham El-Rewini Mostafa Abd-El-Barr. *ADVANCED COMPUTER ARCHITECTURE AND PARALLEL PROCESSING*. John Wiley, 2005
21. Hwang, K. and Briggs, F. A. *Computer Architecture and Parallel Processing*, McGraw-Hill, 1984.
22. Ibbett, R. N. and Topham, N. P. *Architecture of High Performance Computers II*, Springer-Verlag, 1989.
23. Juang, J.-Y. and Wah, B. A contention-based bus-control scheme for multiprocessor systems. *IEEE Transactions on Computers*, 40 (9), 1046–1053 (1991).
24. Flynn M.F. Some computer organizations and their effectiveness. – *IEEEETC*, 1972, September. P. 848–960.
25. Lewis, T. G. and El-Rewini, H. *Introduction to Parallel Computing*, Prentice-Hall, 1992.
26. Linder, D. and Harden, J. An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes. *IEEE Transactions on Computers*, 40 (1), 2–12 (1991).
27. Moldovan, D. *Parallel Processing, from Applications to Systems*, Morgan Kaufmann Publishers, 1993.
28. Ni, L. and McKinley, P. A survey of wormhole routing techniques in direct networks. *IEEE Computer*, February 1993, 62–76 (1993).
29. Patel, J. Performance of processor–memory interconnections for multiprocessor computer systems. *IEEE Transactions*, 28 (9), 296–304 (1981).
30. Reed, D. and Fujimoto, R. *Multicomputer Networks: Message-Based Parallel Processing*, MIT Press, 1987.
31. Serlin, O. The Serlin Report On Parallel Processing, No. 54, pp. 8–13, November 1991.
32. Sima, E., Fountain, T. and Kacsuk, P. *Advanced Computer Architectures: A Design Space Approach*, Addison-Wesley, 1996.
33. Stone, H. *High-Performance Computer Architecture*, 3rd ed., Addison-Wesley, 1993.
34. The Accelerated Strategic Computing Initiative Report, Lawrence Livermore National Laboratory, 1996.
35. Wilkinson, B. *Computer Architecture: Design and Performance*, 2nd ed., Prentice-Hall, 1996.
36. Yang, Q. and Zaky, S. Communication performance in multiple-bus systems. *IEEE Transactions on Computers*, 37 (7), 848–853 (1988).

37. Youn, H. and Chen, C. A comprehensive performance evaluation of crossbar networks. *IEEE Transactions on Parallel and Distribute Systems*, 4 (5), 481–489 (1993).
38. Zargham, M. *Computer Architecture: Single and Parallel Systems*, Prentice-Hall, 1996.
39. Мельник А. А., Ільків В. С. Реалізація алгоритмов сортировки. Систоліческі вычислительные структуры. Препринт N3-87. ИППММ АН УССР. – Львов, 1988. – с. 25–26.
40. Мельник А. А. О подходе к реализации многоступенчатых коммутирующих сетей. Высокопроизводительные вычислительные системы. Препринт N6-89. – Львов, 1989. – с. 46–47.
41. Мельник А. О. Принципи організації управління для одного класу багатоступінчастих комутуючих мереж. Матеріали НТК “Досвід розробки та застосування приладо-технологічних САПР мікроелектроніки”. – Львів, 1995. – Ч. 1. – с. 28–29.

12.13. Питання до розділу 12

1. Наведіть хронологію нововведень в архітектурі комп’ютера з точки зору паралельної обробки інформації.
2. В якому комп’ютері вперше була використана паралельна пам’ять та паралельний АЛП?
3. В якому комп’ютері вперше використані процесори введення-виведення?
4. Які принципово нові архітектурні рішення були реалізовані в комп’ютері Stretch?
5. В якому комп’ютері вперше був використаний конвеєрний принцип виконання команд, віртуальна пам’ять та система переривань?
6. В комп’ютерах якої серії вперше були використані незалежні конвеєрні операційні пристрії?
7. Наведіть основні архітектурні принципи матричних процесорів та продемонструйте їх застосування на прикладі процесора ILLIAC IV.
8. Дайте оцінку впливу векторно-конвеєрних комп’ютерів фірми CRAY на подальший розвиток комп’ютерів.
9. Виділіть особливості паралельних комп’ютерів з спільною пам’яттю.
10. Наведіть особливості та приклади систем з масовою паралельною обробкою інформації.
11. Наведіть закон Амдала та наслідки, що витікають з цього закону.
12. Яка основна ідея багатопотокової обробки інформації?
13. Які існують основні підходи до реалізації багатопотокової обробки інформації?
14. Опишіть технологію Hyper-Threading, використану в процесорі Xeon фірми Intel.
15. Наведіть основні положення класифікації комп’ютерних систем, запропоновані Шором.
16. Наведіть структуру машини 1 за класифікацією Шора.
17. Наведіть структуру машини 2 за класифікацією Шора.
18. Наведіть структуру машини 3 за класифікацією Шора.
19. Наведіть структуру машини 4 за класифікацією Шора.
20. Наведіть структуру машини 5 за класифікацією Шора.
21. Наведіть структуру машини 6 за класифікацією Шора.
22. Наведіть принципи класифікації комп’ютерних систем, запропоновані Фліном.
23. Наведіть структуру комп’ютерної системи ОКОД.
24. Наведіть структуру комп’ютерної системи МКОД.
25. Наведіть структуру комп’ютерної системи ОКМД.
26. Наведіть структуру комп’ютерної системи МКМД.
27. Наведіть структуру багатопроцесорної системи типу ОКМД з розподіленою пам’яттю.
28. Наведіть структуру багатопроцесорної системи типу ОКМД з спільною пам’яттю.
29. Поясніть структуру та організацію роботи багатопроцесорної системи типу МКМД з спільною пам’яттю.

30. Поясніть структуру та організацію роботи багатопроцесорної системи типу МКМД з розподіленою пам'яттю.
31. Наведіть класифікацію багатопроцесорних систем типу МКМД з спільною пам'яттю.
32. Поясніть основні особливості комп'ютерних систем з однорідним доступом до пам'яті.
33. Поясніть основні особливості комп'ютерних систем з неоднорідним доступом до пам'яті.
34. Поясніть основні особливості комп'ютерних систем лише з кеш пам'яттю.
35. Поясніть загальну організацію роботи комп'ютерних систем з розподіленою пам'яттю.
36. Наведіть типи комутаційних мереж багатопроцесорних систем.
37. Які існують статичні топології комутаційних мереж багатопроцесорних систем.
38. Наведіть особливості та основні характеристики комутаційних мереж з повним з'єднанням.
39. Наведіть особливості та основні характеристики комутаційних мереж з неповним з'єднанням.
40. Порівняйте комунікаційні мережі з статичним з'єднанням – 1-вимірні, 2-вимірні, 3-вимірні.
41. Наведіть основні типи шинних комутаційних мереж багатопроцесорних систем та порівняйте їх між собою.
42. Як здійснюється синхронізація шин шинних комутаційних мереж?
43. Як працює комунікаційна мережа типу Crossbar?
44. Наведіть структури одноярусних комутаційних мереж.
45. Наведіть структури багатоярусних комутаційних мереж: "Баньян", "Омега".
46. Наведіть структури комутаційних мереж з реконфігурацією.
47. Наведіть структури неблокуючих комутаційних мереж.

Наукове видання

МЕЛЬНИК А. О.

*Архітектура
комп'ютера*

Головний редактор – Дмитро Головенко

В авторській редакції

Технічний редактор – Віра Озірська

Обкладинка – Володимир Оверчук

Комп'ютерна верстка – Оксана Мисюк

Коректор – Надія Фалюш

Підписано до друку 21.05.2008 р. Формат 70x100 1/16.

Папір офсетний. Гарнітура Minion Pro.

Офсетний друк. Обл.-вид. арк. 44,1. Ум. друк. арк. 38,2.

Тираж 3000. Зам. 371.

Видавництво “Волинська обласна друкарня”

43010 м. Луцьк, пр. Волі, 27.

Тел.: 4-25-01, 4-25-07, 4-41-73.

Свідоцтво Держкомінформу України ДК № 1350 від 13.05.2003 р.

Друк та палітурні роботи ВАТ “Волинська обласна друкарня”

43010 м. Луцьк, пр. Волі, 27.

Тел.: 4-25-01, 4-25-07, 4-41-73.

Мельник А. О.

М 48 Архітектура комп'ютера. Наукове видання. – Луцьк: Волинська обласна друкарня, 2008. – 470 с.

ISBN 978-966-361-264-5

Підручник охоплює весь комплекс питань, пов'язаних з теорією, принципами та методами побудови та організації функціонування комп'ютерів. Розкриті питання представлення команд та даних в комп'ютері, алгоритми виконання операцій обробки даних та принципи побудови арифметико-логічного пристрою. Описана структура та організація роботи комп'ютера з простою та складною системою команд, суперконвеєрного, суперскалярного, векторного та з довгим форматом команди. Розглянуті конфлікти в конвеєрі команд та методи їх усунення, структура та організація роботи пристрою керування. Наведена структура пам'яті комп'ютера, організація взаємодії між її рівнями, сегментна організація пам'яті та питання її захисту. Дане пояснення способів розпізнавання пристрій введення-виведення та організація їх роботи. Розглянуті питання подальшого підвищення продуктивності комп'ютера та шляхи створення паралельних комп'ютерних систем.

Підручник призначений для студентів вищих навчальних закладів України. Він також може бути корисним для викладачів, фахівців в області комп'ютерних та інформаційних технологій.

УДК 004.2

ББК 32.973



Мельник Анатолій Олексійович, доктор технічних наук, професор, завідувач кафедри електронних обчислювальних машин Національного університету «Львівська політехніка», декан факультету комп’ютерних та інформаційних технологій Інституту підприємництва і перспективних технологій, директор НВП «Інтрон», основний розробник, науковий керівник та головний конструктор комп’ютерних пристріїв та систем більше 70 наукових проектів, виконаних в наукових установах України та за кордоном, автор понад 300 наукових праць, основний напрямок наукових досліджень – створення комп’ютерів і комп’ютерних систем, розробка теоретичних основ їх побудови та методів проектування. Членство в професійних організаціях: МАКНС, АІН, IEE, IEEE, ACM.

Фундаментальний курс, який поєднує широту охоплення матеріалу з детальним розглядом особливостей архітектури комп’ютера.

Інформація, яку ви знайдете в книзі:

- історичні аспекти розвитку комп’ютерів
- основні елементи архітектури комп’ютера
- представлення даних у комп’ютері
- процесор
- запобігання конфліктам у конвеєрі команд
- алгоритми виконання операцій обробки даних
- арифметико-логічний пристрій
- пристрій керування
- багаторівнева пам’ять
- ієрархічна організація пам’яті
- пристрій введення-виведення
- паралельні комп’ютерні системи

Підручник орієнтований на студентів вищих навчальних закладів, аспірантів, викладачів та науково-технічних спеціалістів, які працюють в галузі комп’ютерних та інформаційних технологій.

ISBN 978-966-361-264-5

9789663612645 30608