

Циганок Віталій Володимирович,

д.т.н., с.н.с.

«Методи та системи паралельного програмування »

галузь знань

"Інформатика та обчислювальна техніка"
напрямок підготовки **"Комп'ютерні науки"**
освітньо-кваліфікаційного рівня **«бакалавр»**

Структура навчальної дисципліни

Загальний обсяг **120 год.**,

в тому числі:

Лекцій – **24 год.**

Лабораторні заняття – **34 год.**

Самостійна робота - **62 год.**

Література

Основна:

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.
2. Эхтер Ш., Робертс Д. Многоядерное программирование. – СПб.: Питер, 2010. – 316 с.
3. Rauber T., G. Runger, Parallel Programming: for Multicore and Cluster Systems. Springer, 2010, 450 P.
4. Mattson T., Sanders B., Massingil Berna L. Patterns for Parallel Programming. Addison-Wesley, 2004, 355 P.
5. Дорошенко А.Е. Математические модели и методы организации высокопроизводительных параллельных вычислений. - К., "Наукова думка", 2000.- 177 с.
6. Корнеев В. Параллельные вычислительные системы.–М. Ноулидж. – 1999.

Література

Додаткова:

1. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. Учебное пособие. – Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2003. – 184 с.
2. Эндрюс, Грегори Р. Основы многопоточного, параллельного и распределенного программирования.: Пер. с англ. – М.: Издательский дом "Вильямс", 2003. – 505 с.
3. Дорошенко А.Ю. Лекції з паралельних обчислювальних систем. Київ: Видавничий дім "КМ Академія", 2003. – 42 с.
4. Немнюгин С.А., Стесик О.Л. Параллельное программирование для многопроцессорных вычислительных систем. – СПб.: БХВ-Петербург, 2002. – 400 с.
5. Introduction to OpenMP (<http://ci-tutor.ncsa.uiuc.edu/login.php>).
6. Introduction to MPI (<http://ci-tutor.ncsa.uiuc.edu/login.php>).
7. MPI Forum (<http://www.mpi-forum.org/>)
8. <http://parallel.ru/>

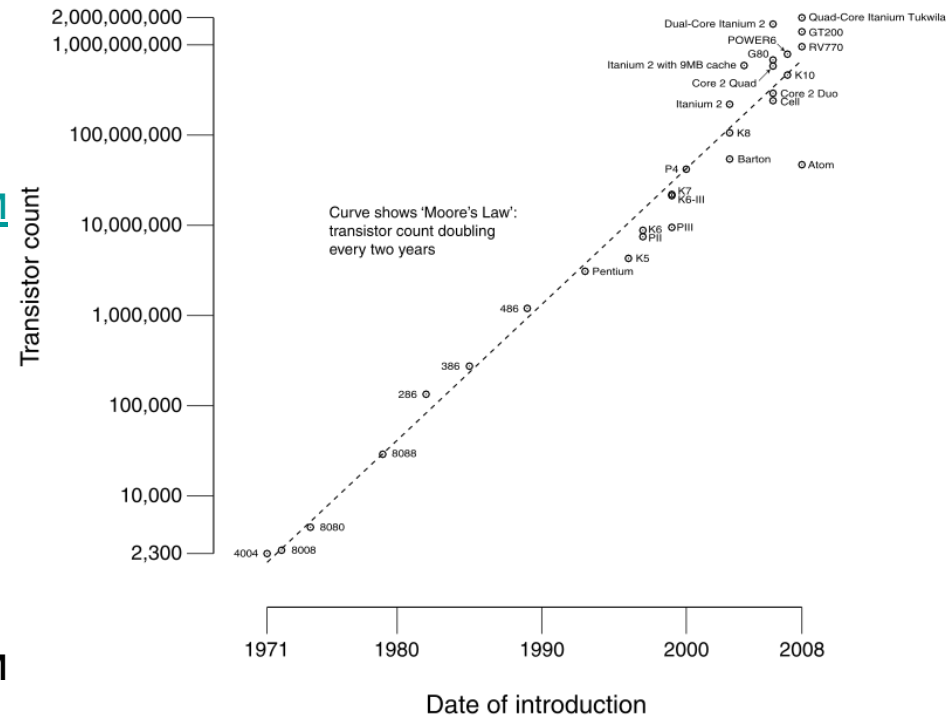
Лекція №1: Головні складові архітектури високопродуктивних обчислювальних систем

1. Історія розвитку високопродуктивних обчислювальних систем.
2. Класифікація паралельних архітектур по Флінну
3. Основні елементи архітектури високопродуктивних обчислювальних систем.
 - 3.1. Конвеєри
 - 3.2. RISC-процесори
 - 3.3. Суперскалярні процесори
 - 3.4. Суперскалярні процесори зі наддовгим командним словом

Закон Мура

- емпіричне— емпіричне спостереження, зроблене в 1965— емпіричне спостереження, зроблене в 1965 році (через шість років після винаходу інтегральної схеми— емпіричне спостереження, зроблене в 1965 році (через шість років після винаходу інтегральної схеми), у процесі підготовки виступу Гордоном Муром— емпіричне спостереження, зроблене в 1965 році (через шість років після винаходу інтегральної схеми), у процесі підготовки виступу Гордоном Муром (одним із засновників компанії Intel— емпіричне спостереження, зроблене в 1965 році (через шість років після винаходу інтегральної схеми), у процесі підготовки виступу Гордоном Муром (одним із засновників компанії Intel). Він висловив припущення, що кількість транзисторів— емпіричне спостереження, зроблене в 1965 році (через шість років після винаходу інтегральної схеми), у

CPU Transistor Counts 1971-2008 & Moore's Law



Паралельні обчислювальні системи

Головне призначення паралельних систем – швидке розв'язання задач. Якщо технологія програмування не дозволяє використовувати весь потенціал обчислювальної системи, використання її втрачає сенс.

Різноманітність архітектур паралельних ЕОМ зумовила різноманітність підходів до їх програмування.

Методи паралельного програмування

```
graph TD; A[Методи паралельного програмування] --> B[Методи]; B --> C[Паралельні мови програмування]; B --> D[Бібліотеки паралельного програмування]; B --> E[Розпаралелюючий компілятор];
```

Методи

**Паралельні
мови
програмування**

**Бібліотеки
паралельного
програмування**

**Розпаралелююч
ий
компілятор**

Порівняння методів паралельного програмування

Паралельні мови програмування	Бібліотеки паралельного програмування	Розпаралелюючий компілятор
<i>Переваги:</i> прозорість коду; порівняно короткі програми; концепція паралелізму прихована в самій мові.	<i>Переваги:</i> можливість використовувати послідовні мови; простота написання програм.	<i>Переваги:</i> можливість розпаралелювання вже існуючих програм; відсутність втручання людини процес розпаралелювання; непотрібність навчання.
<i>Недоліки:</i> необхідність вивчення правил написання паралельних програм; необхідність видозміни алгоритмів.	<i>Недоліки:</i> прив'язаність бібліотеки до конкретної платформи; громіздкість коду; необхідність контролю потоків і синхронізації.	<i>Недоліки:</i> складність реалізації; складність досягнення оптимальної продуктивності.
<i>Використання:</i> Написання невеликих ефективних програм.	<i>Використання:</i> Швидке написання паралельних програм.	<i>Використання:</i> Перенесення послідовних програм на паралельні архітектури.

Історія розвитку високопродуктивних обчислювальних систем

70-ті роки ХХст.

- - Експериментальні розробки по створенню багатопроцесорних обчислювальних систем (БОС)(розроблена в Іллінойському університеті ILLIAC IV, включала 64 (у проекті до 256) процесорних елементи (ПЕ), що працюють за єдиною програмою, застосовуваної до вмісту власної оперативної пам'яті кожного ПЕ. Обмін даними між процесорами здійснювався через спеціальну матрицю комунікаційних каналів => Назва "матричні суперкомп'ютери")
- - БОС з розподіленою пам'яттю і з довільною комунікаційною системою отримав згодом назву "багатопроцесорні системи з масовим паралелізмом", або БОС з MPP- архітектурою (MPP - MassivelyParallelProcessing). як правило, кожен з ПЕ MPP системи є універсальним процесором, що працює за своєю власною програмою.

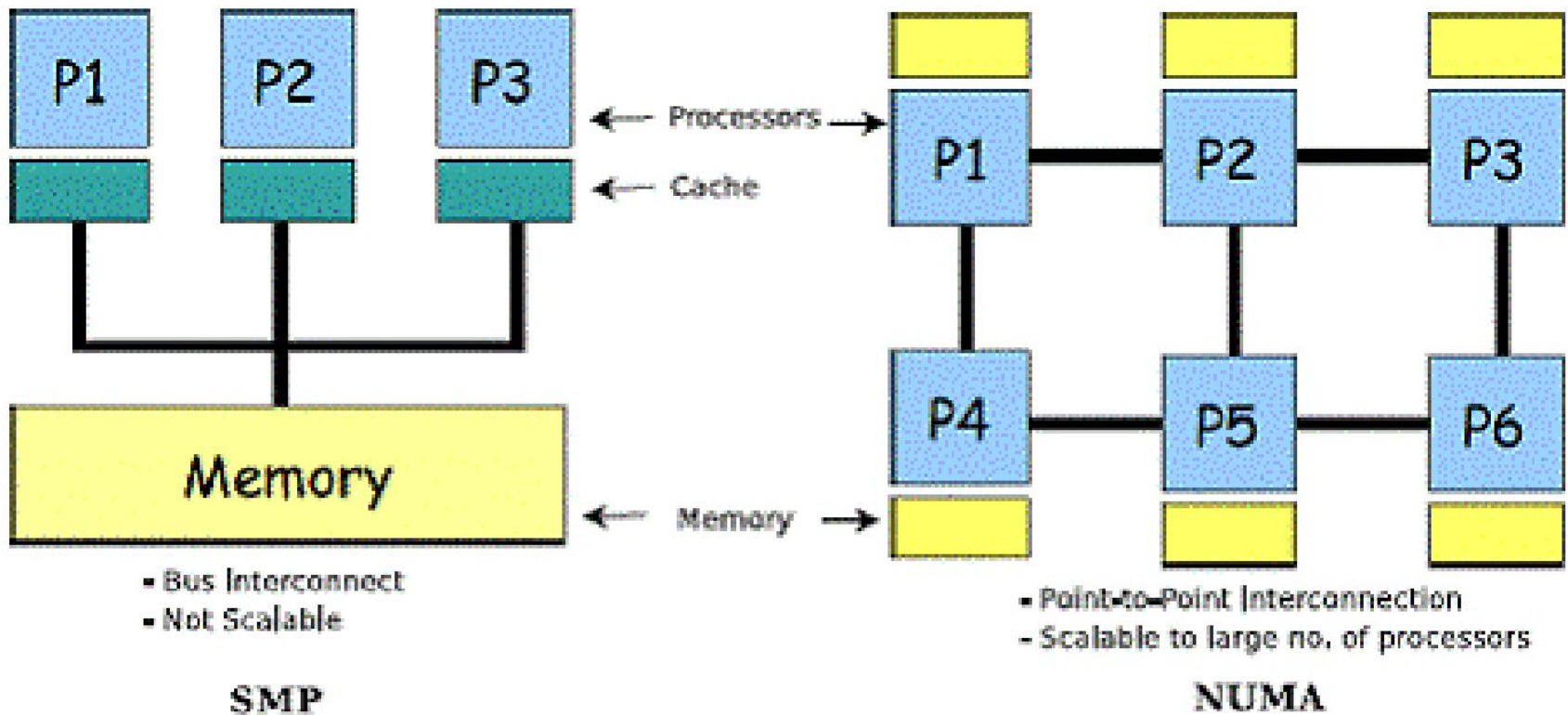
Історія розвитку високопродуктивних обчислювальних систем

80-ті роки ХХст.

- Перші промислові зразки мультипроцесорних систем на базі векторно-конвеєрних комп'ютерів (суперкомп'ютери фірми **Cray**: 2-16 процесорів з рівноправним (*симетричним*) доступом до загальної оперативної пам'яті => назва симетричні мультипроцесорні системи (Symmetric Multi-Processing - SMP).
- Ідея забезпечити кожен процесор власною оперативною пам'яттю, перетворюючи комп'ютер у об'єднання незалежних обчислювальних вузлів. + розробка спеціального способу обміну даними між обчислювальними вузлами, реалізованого зазвичай у вигляді механізму передачі повідомлень (Message Passing).

SMP і NUMA-архітектури

(NonUniformMemoryAccess) - середнє між SMP і MPP. В NUMA-архітектурах пам'ять фізично розділена, але логічно загальнодоступна.



Сучасний розвиток суперкомп'ютерних технологій

4 основні напрямки:

1. **векторно-конвеєрні суперкомп'ютери**
2. **SMP системи**
3. **MPP системи і**
4. **Кластерні системи**

Класифікація паралельних архітектур по Флінну

запропонована Майклом Флінном в 1966 році і розширена в 1972 році



Класифікація паралельних архітектур по Флінну

- ОКОД - Обчислювальна система з одиночним потоком команд і одиночним потоком даних (**SISD**, Single Instruction stream over a Single Data stream).
- ОКМД - Обчислювальна система з одиночним потоком команд і множинним потоком даних (**SIMD**, Single Instruction, Multiple Data).
- МКОД - Обчислювальна система з множинним потоком команд і одиночним потоком даних (**MISD**, Multiple Instruction Single Data).
- МКМД - Обчислювальна система з множинним потоком команд і множинним потоком даних (**MIMD**, Multiple Instruction Multiple Data).

Схема SISD - комп'ютера



Схема SIMD - комп'ютера з роздільною пам'яттю



Схема SIMD - комп'ютера з розподіленою пам'яттю

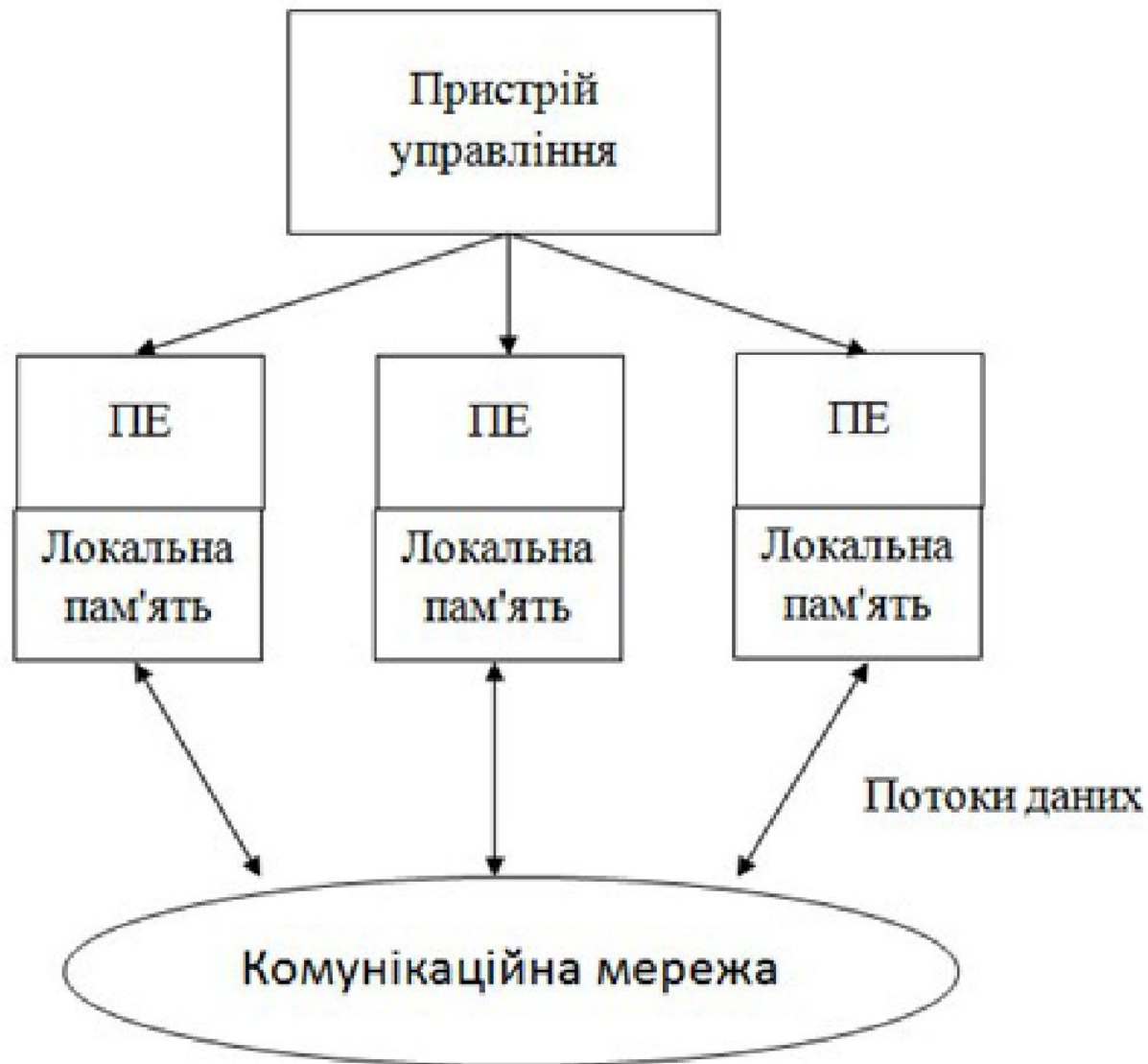


Схема MISD - комп'ютера

Клас MISD включає в себе багатопроцесорні системи, де процесори обробляють множинні потоки даних.

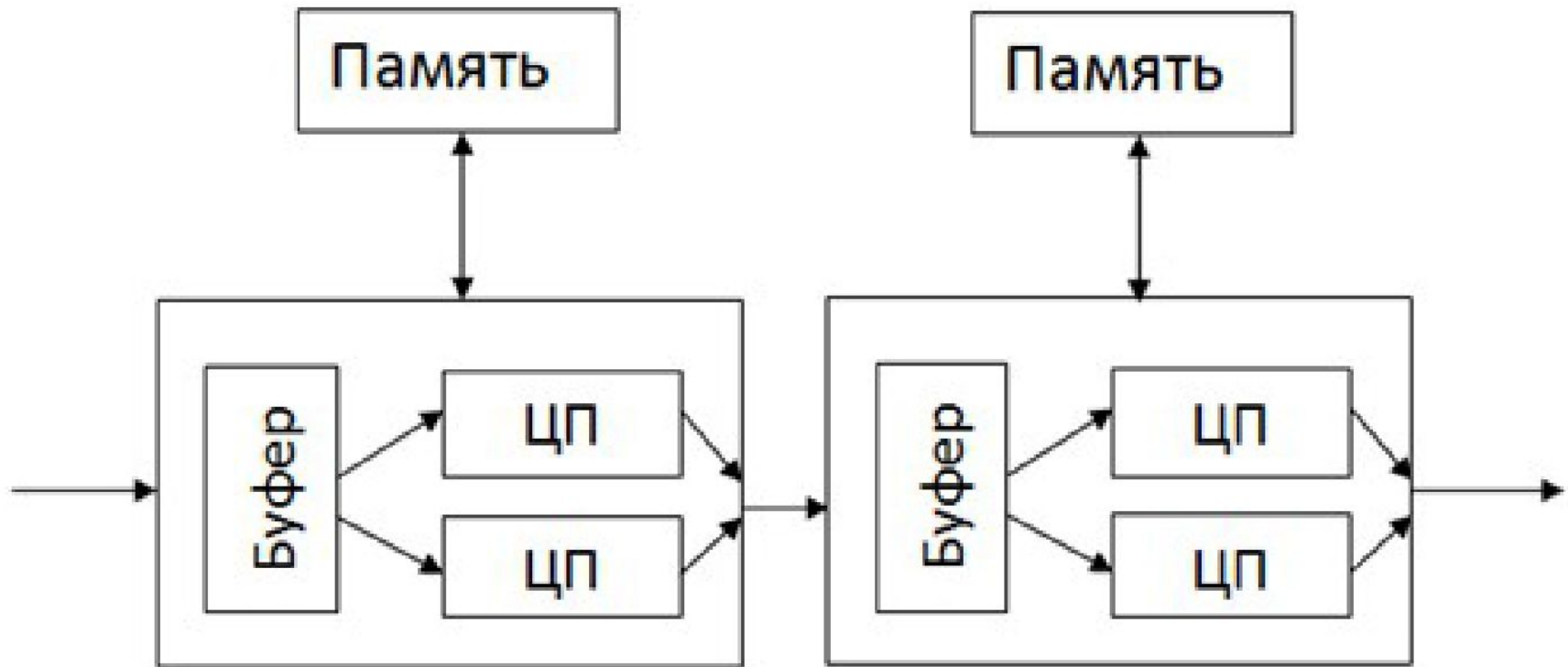


Схема MIMD - комп'ютера з роздільною пам'яттю

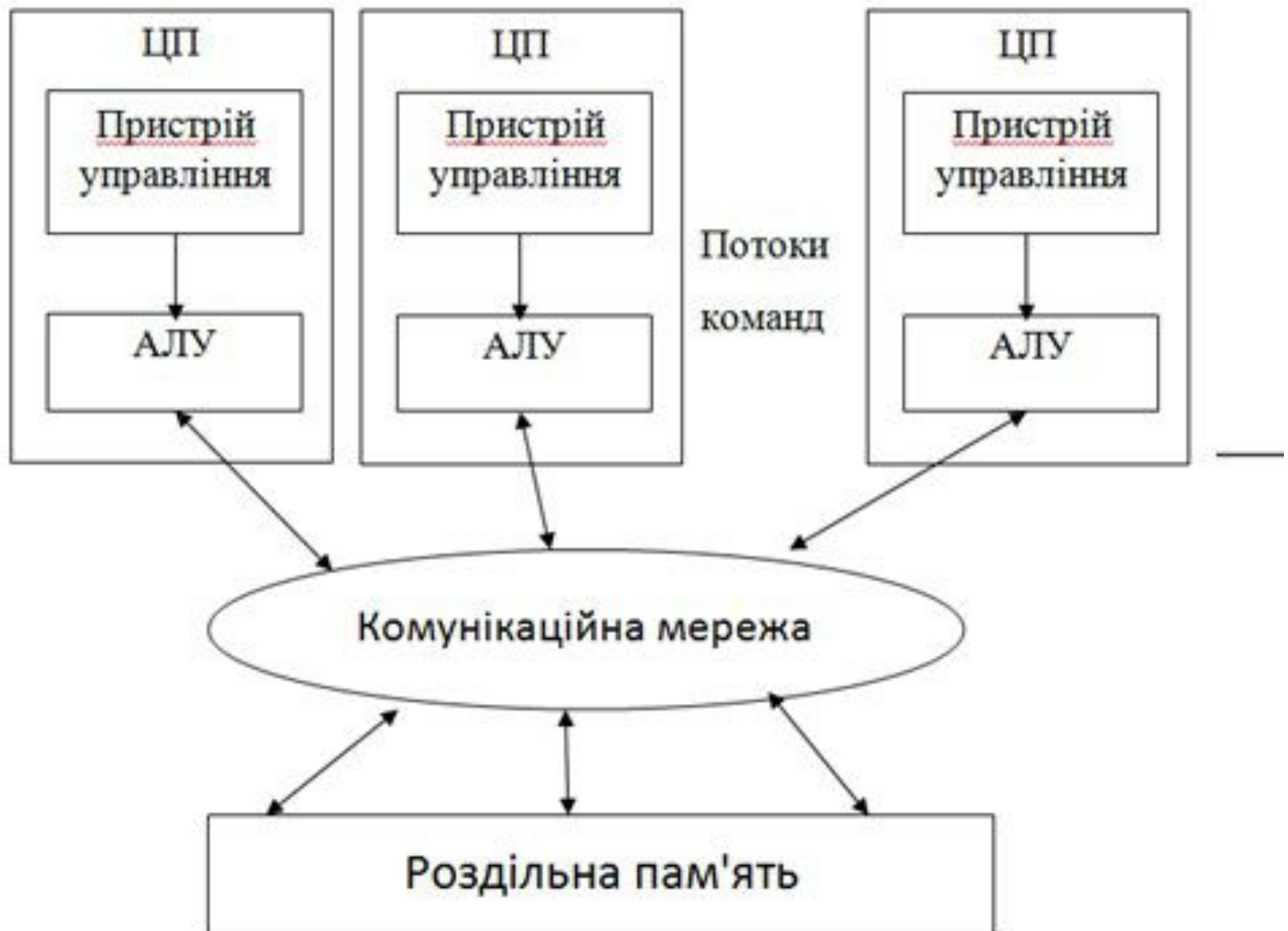
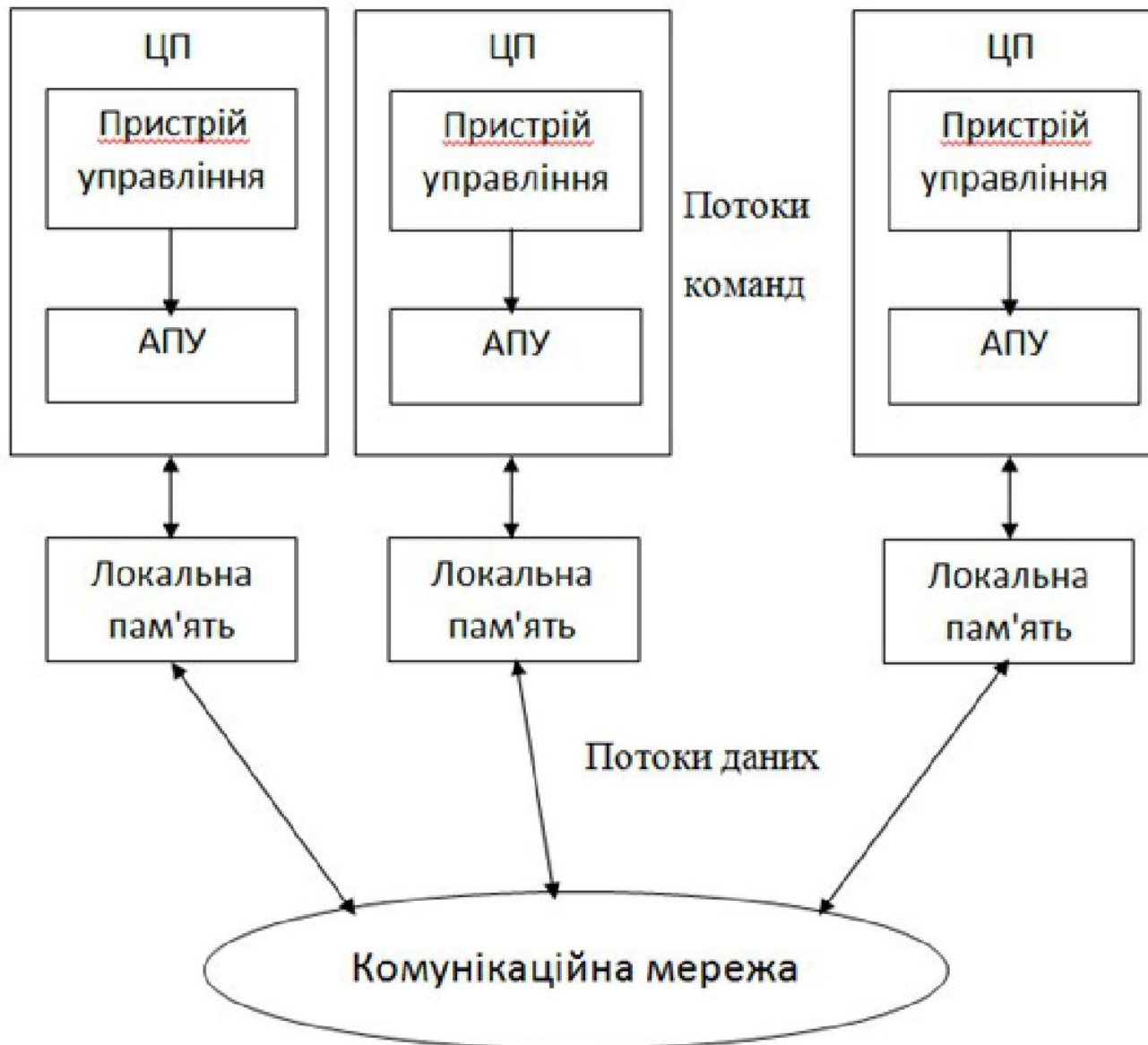


Схема MIMD - комп'ютера з розподіленою пам'яттю



Основні елементи архітектури високопродуктивних обчислювальних систем

1. Конвеєри
2. RISC-процесори
3. Суперскалярні процесори
4. Суперскалярні процесори із
наддовгим командним словом

Конвеєри

Ідея конвеєра полягає в тому, щоб складну операцію розбити на кілька більш простих, таких, які можуть виконуватися одночасно.

Для ефективної реалізації конвеєра повинні виконуватися наступні умови:

- система виконує повторювану операцію;
- операція може бути розділена на незалежні частини;
- трудомісткість підоперацій приблизно однакова.



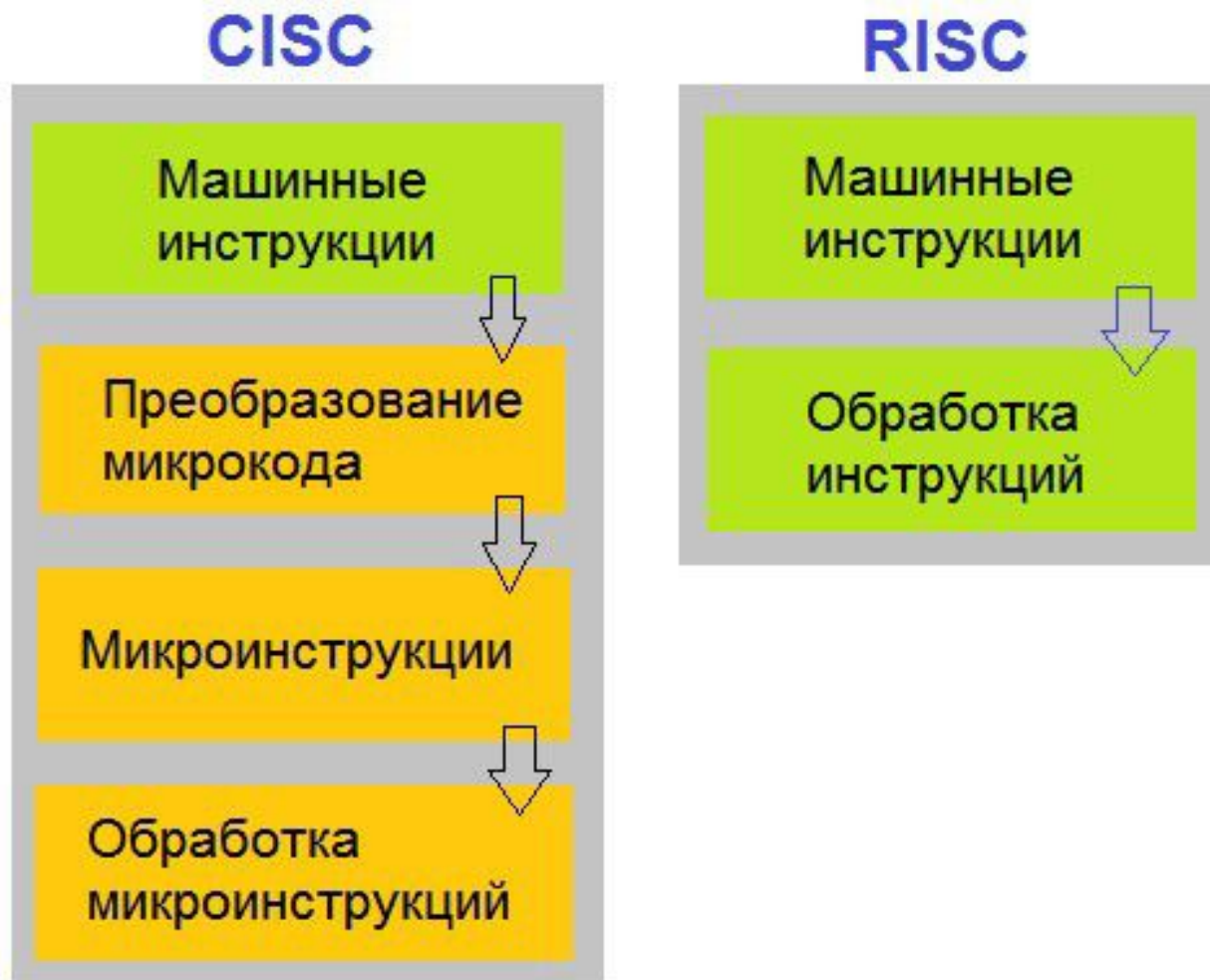
RISC- та CISC- процесори

В основі RISC-архітектури (RISC - *Reduced Instruction Set Computer*)

процесора лежить ідея збільшення швидкості його роботи за рахунок спрощення набору команд.

Протилежну тенденцію представляють CISC-архітектури, процесори зі складним набором команд (CISC - *Complete Instruction Set Computer*).

Порівняння CISC і RISC процесорів



Суперскалярні процесори

У них реалізована конвеєрна обробка і паралельне виконання команд.

У звичайному конвеєрі паралельне виконання команд можливо тільки при знаходженні команд на різних стадіях обробки (у різних сегментах конвеєра).

Суперскалярні процесори дозволяють виконувати декілька команд в одному сегменті конвеєра. *Кілька команд одночасно можуть виконатися протягом одного такту.* У суперскалярних процесорах використовується **декілька конвеєрів, що працюють паралельно**, і **пристрої для інтерпретації команд, забезпечені логікою, що дозволяє визначити, чи є команди незалежними.**

Представники: Pentium, PowerPC, K6 / K7, Alpha

Суперскалярні процесори із наддовгим командним словом

Суперскалярні процесори зі наддовгим командним словом – **VLIW** (Very Long Instruction Word) в процесі роботи виявляють паралелізм команд під час трансляції.

Транслятор, аналізуючи програму, *виділяє операції, які можна виконувати паралельно, і будує з них великі команди.* Як тільки велика команда надходить на виконання, звичайні команди з неї виконуються паралельно.

Така реалізація веде до ряду **недоліків**:

- немає сумісності на рівні бінарних файлів,
- необхідний канал з великою розрядністю,
- збільшується виконуваний код).

Схема процесора з наддовгим командним словом

