

Тема 2. Класифікація архітектур ОС

Для ЕОМ першого і другого покоління характерною була класифікація за системою команд. Практично перехід від однієї системи до іншої означав повну перекваліфікацію програміста. У ЕОМ третього і четвертого покоління щораз більше застосовували мови програмування високого рівня (Фортран, ПЛ1, Алгол та ін.), і програмісти головно не потребували знання системи команд, структури пам'яті, способів адресування, розрядності регістрів тощо. Однак для системного програміста і сьогодні актуальними є питання структури ЕОМ, ємність пам'яті і взаємодії між компонентами, особливо між процесором і периферією, тому, погоджуючись зі А. Смирновим [1], розглянемо таку класифікацію.

2.1. Класифікація за інтегральними ознаками

Доцільно скористатися поняттям архітектури ЕОМ як певним набором інтегральних характеристик, а саме:

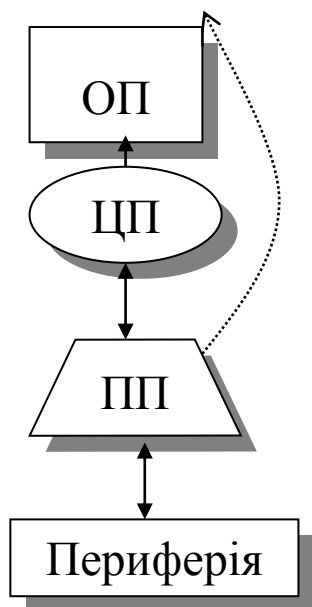
- сукупність властивостей і характеристик, що визначають інформаційні зв'язки між ЦП, периферійними процесорами (ПП), ОП, зовнішньою пам'яттю і периферією;
- взаємодією і структурою потоків команд та потоків даних.

Перша характеристика дає змогу врахувати такі особливості ЕОМ третього і четвертого поколінь, як багатопроцесорність, моношинна структура зв'язків, швидкодія, розподілена пам'ять.

2.1.1. Класифікація за взаємодією ЦП, ОП, ПП

Розглянемо інформаційні зв'язки між трьома головними типами пристроїв ЕОМ, обмежившись лише структурними зв'язками в архітектурі ОС. Є такі структури:

- однопроцесорна (з одним ЦП і периферійним процесором чи каналом) зі зв'язком через ЦП;
- з одним потужним ЦП і кількома периферійними процесорами, що його обслуговують;
- багатопроцесорна (кілька ЦП, ПП і секціонована ОП);
- з магістральною шиною;
- мережна;
- функціонально переналагоджувальна багатопроцесорна;
- мас-процесорна (дуже значна кількість процесорів).



В однопроцесорній структурі ЦП (спільно з операційною системою) координує всю роботу системи. Потрібні прямі зв'язки з периферійними пристроями ЦП виконує через канали або ПП (рис. 2. 1). Така структура була характерною для ЕОМ третього покоління: IBM 360/370, ЕС ЕОМ, ICL System4 та інші. Ця архітектура застосовується також в сучасних пристроях, які забезпечують виконання спеціальних функцій у комп'ютерних мережах.

Рис. 2. 1. Однопроцесорна архітектура ЕОМ

Однак для продуктивніших систем такі структури, коли в разі кожного звертання необхідно переривати роботу ЦП, не вигідні. Тому виникла ідея поєднати роботу потужного ЦП із кількома менш потуж-

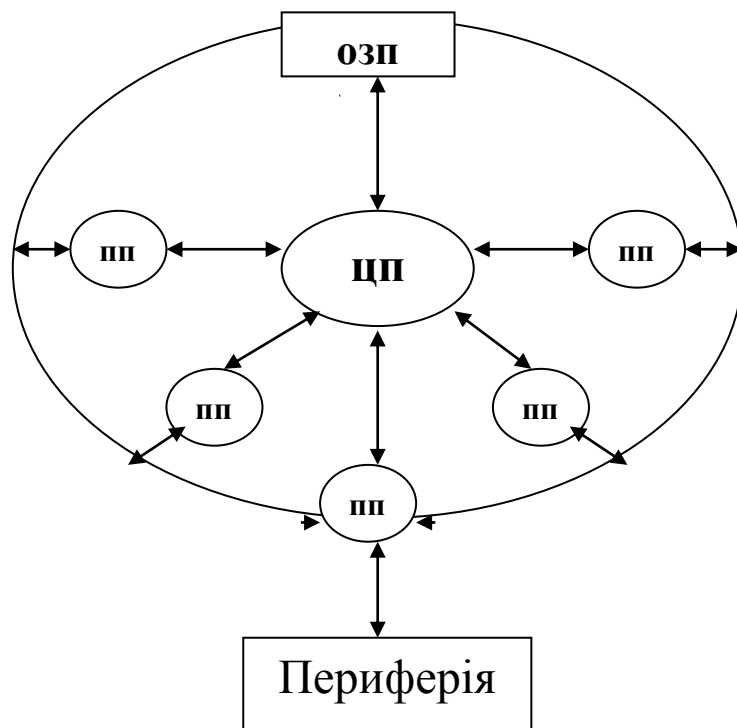


Рис. 2. 2. Архітектура з периферійними процесорами

нішими, функція яких полягає у обслуговуванні переривань від зовнішніх пристроїв, а також ОЗП. Наприклад, це може бути структура з концентричним розташуванням периферійних процесорів (рис. 2.2). Подібну структуру мають високопродуктивні системи CDC.

А от ЕОМ "Ельбрус" (кількість ЦП від 1 до 10, ПП від 1 до 4, кількість секцій ОП від 1 до 8), яку донедавна використовували тільки для військових цілей, була побудована за принципом збільшення кількості ЦП з однією ОП. Система може працювати за алгоритмом розпаралелювання задачі на кілька процесів або розв'язувати на кожному з них свою задачу. Ця система застосована також у Burroughs-700 (США), Iris-80 (Франція) та ін. (рис. 2. 3).

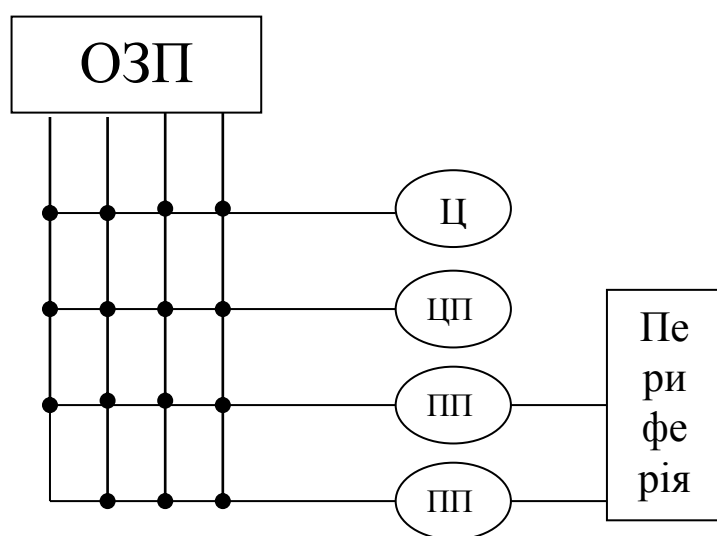


Рис. 2. 3. Багатопроцесорна архітектура ЕОМ

Збільшення продуктивності такої системи близьке до лінійного до трипроцесорного варіанта (рис. 2. 4). Тому серед ОС високої продуктивності популярні дво- і чотирипроцесорні системи (Cray-X-MP; IBM "Сьєрра" та ін.). Завдяки використанню комплексу додаткових засобів і архітектурних знахідок з опорою на кеш-пам'ять великої потужності вдалося отримати майже лінійне зростання продуктивності зі збільшенням кількості процесорів до п'яти-десяти для широкого класу задач на ОС родини "Ельбрус".

Нові можливості відкриваються для паралельних ОС, якщо звернутися до різного типу сіткових алгоритмів. Такими алгоритмами розв'язують багато задач неперервного середовища, що їх описують системи рівнянь у частинних похідних, лінійні системи алгебричних рівнянь. Процесори, які опрацьовують інформацію та

Продуктивність,
умовних одиниць

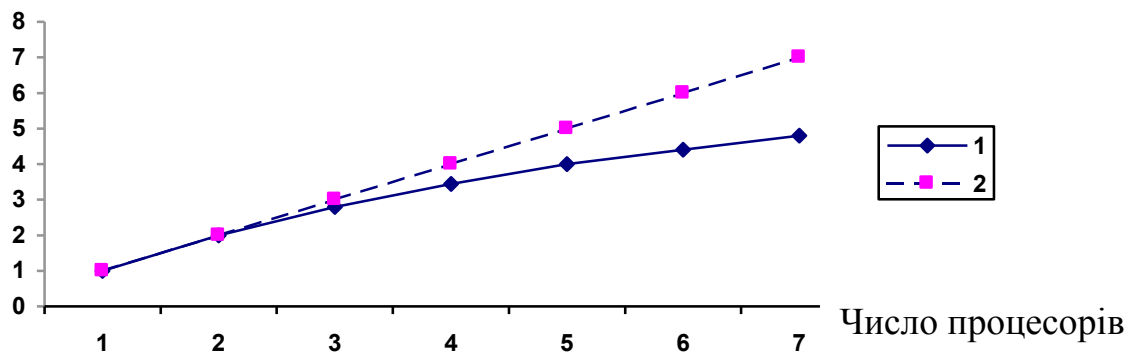


Рис.2.4. Залежність продуктивності ОС від числа процесорів:
1 – реальна, 2 – очікувана.

обмінюються інформацією між собою, розміщені у вузлах решітки або сітки. Таких процесорів у ОС міститься вже декілька сотень або навіть тисяч. Основою ОС стають мікропроцесори. На відміну від багатопроесорних систем з великими універсальними процесорами, такі системи називають *мас-процесорними*.

Операції розподіляють за окремими функціональними пристроями, які працюють паралельно, проте в самих функціональних пристроях застосована конвеєризація (див. стор. 35), а також відбувається попередній перегляд програм, а їхньому швидкому попередньому і паралельному виклику з ОП допомагає глибоке розшарування пам'яті. Детальніше про це див. напр., у [1].

Розрізняють такі мас-процесорні ОС:

- багатопроесорні скалярні (IBM 3090/200, ЕС 1065);
- векторні і векторно-конвеєрні (CDC 6600, Cray -1);
- мас - процесорні з динамічною архітектурою;
- спеціальні процесорні.

Про *магістральну шину* детальніше йтиметься під час розгляду архітектури мікропроцесорних систем. Тут зазначимо лише таке: магістрально-шинна структура передбачає, що всі частини ОС можуть бути приєднаними до однієї багатопровідної шини (або коаксіального чи оптоволоконного кабелю), захоплюючи її на короткий період машинного такту для обміну інформацією (рис. 2. 5). Конфліктну ситуацію вирішує арбітр шини.

Розрізняють шину адреси, шину даних та шину керування.

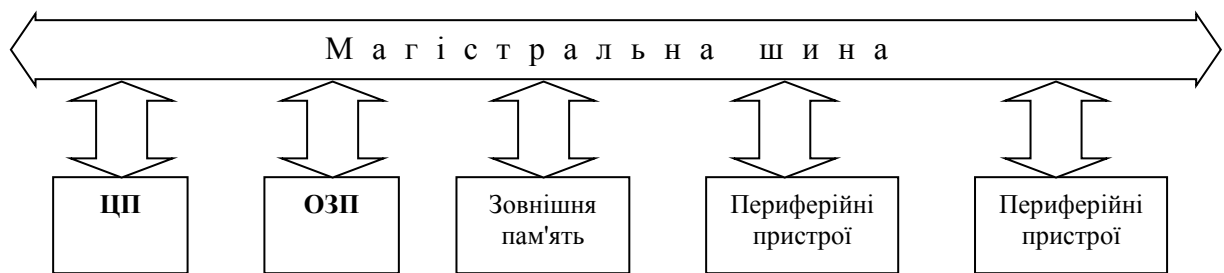


Рис. 2.5. Приклад архітектури ОС з магістральною шиною

Окремий напрям становить мережна архітектура ОС. Розрізняють глобальні, регіональні та локальні комп'ютерні мережі. Вони мають свою топологію і їхня робота регламентована певним набором правил (протоколів). Про це детальніше див. напр. у [4,5].

2.1.2. Класифікація за взаємодією потоку команд і потоку даних

Розглянемо дещо іншу класифікацію, яка властива головно суперЕОМ. Відомо, що у ЕОМ з класичною архітектурою опрацювання даних є послідовним. Команди надходять одна за одною, а для них з ОП чи з регістрів так само послідовно надходять операнди. Одній команді відповідає один необхідний для неї набір операндів; для бінарних операцій, як звичайно – два набори. Скорочено такий тип архітектури називають ОКОД (одна команда – одні дані); англійська аббревіатура SISD (Single Instruction, Single Data) (рис. 2. 6).

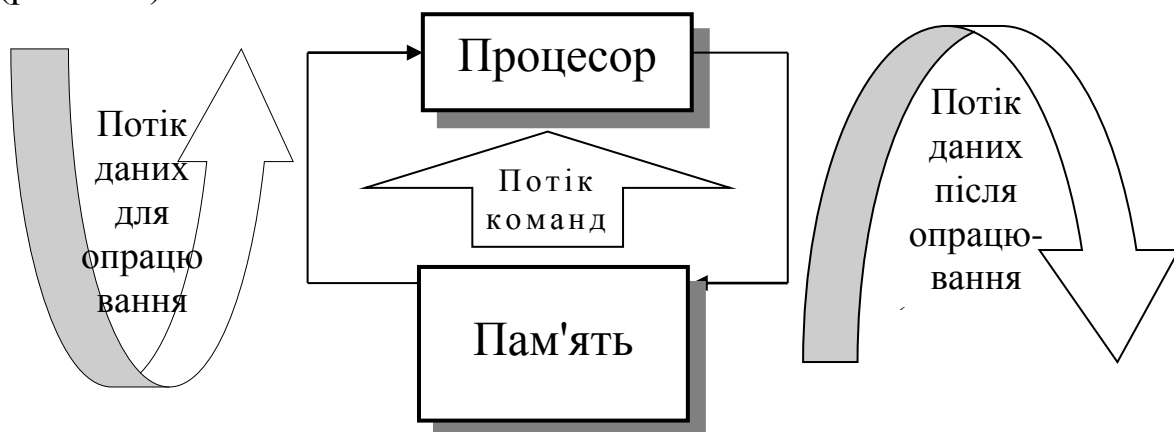


Рис.2.6. Тип архітектури ОКОД

Для спеціальних процесорів, що опрацьовують зображення, або асоціативних процесорів опрацьовується набір з множини даних. Це тип ОКБД (одна команда – багато даних); англійська аббревіатура SIMD (Single Instruction, Multiple Data) (рис. 2.7). Кількість елементів, що опрацьовуються, може бути дуже великою (тисячі), проте їх опрацювання відбувається синхронно (приклад: сучасні процесори з технологією MMX).

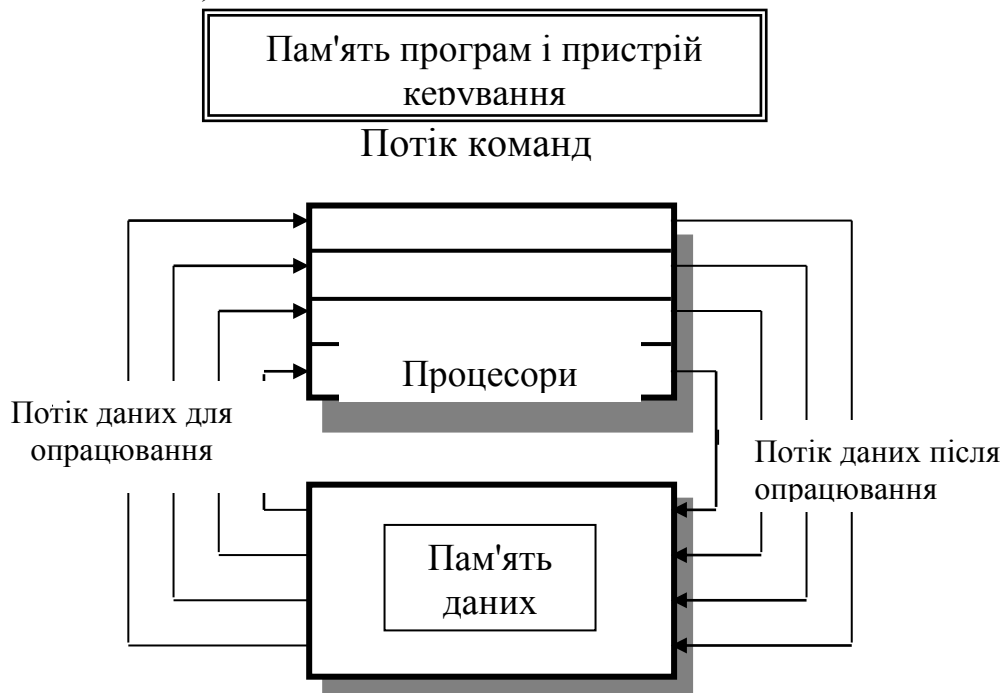


Рис. 2.7. Тип архітектури ОКБД

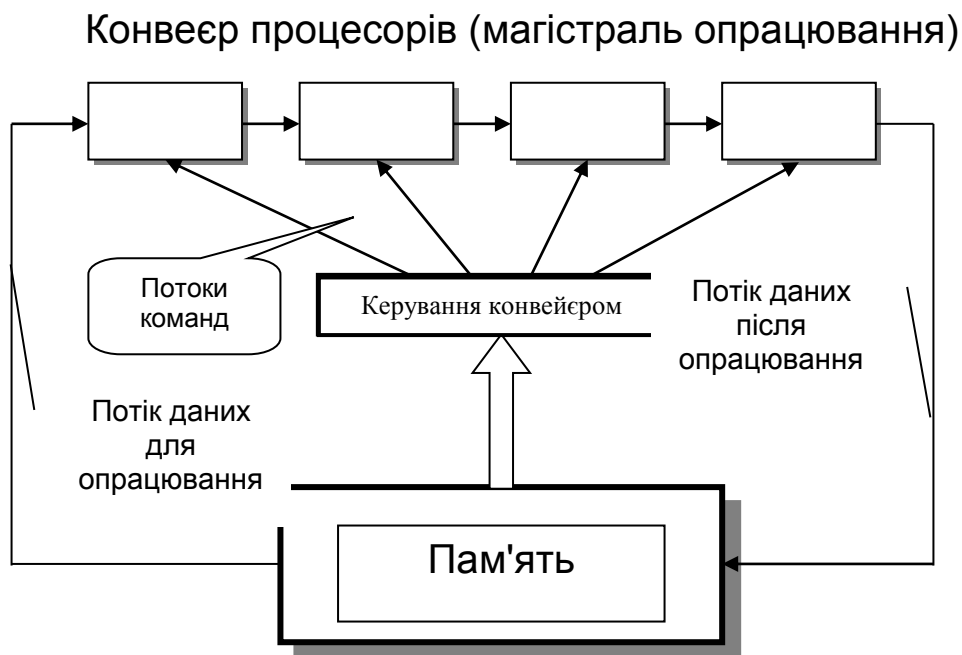


Рис. 2. 8. Тип архітектури БКОД

Прикладом типу БКОД (багато команд – одні дані) може бути конвеєр, тобто послідовне опрацювання одного потоку даних багатьма опрацьовувальними ступенями конвеєра (рис.2.8).

Тип БКБД (багато команд – багато даних) передбачає найповніше і незалежне розпаралелювання процесу (рис. 2. 9).

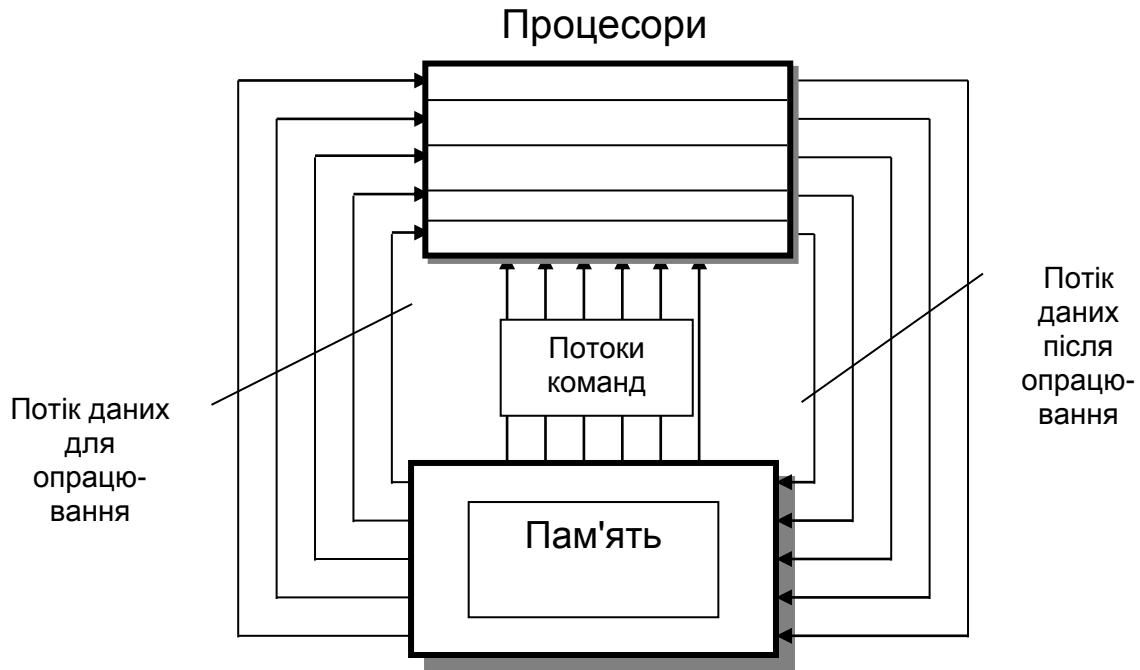


Рис. 2. 9. Архітектура БКБД

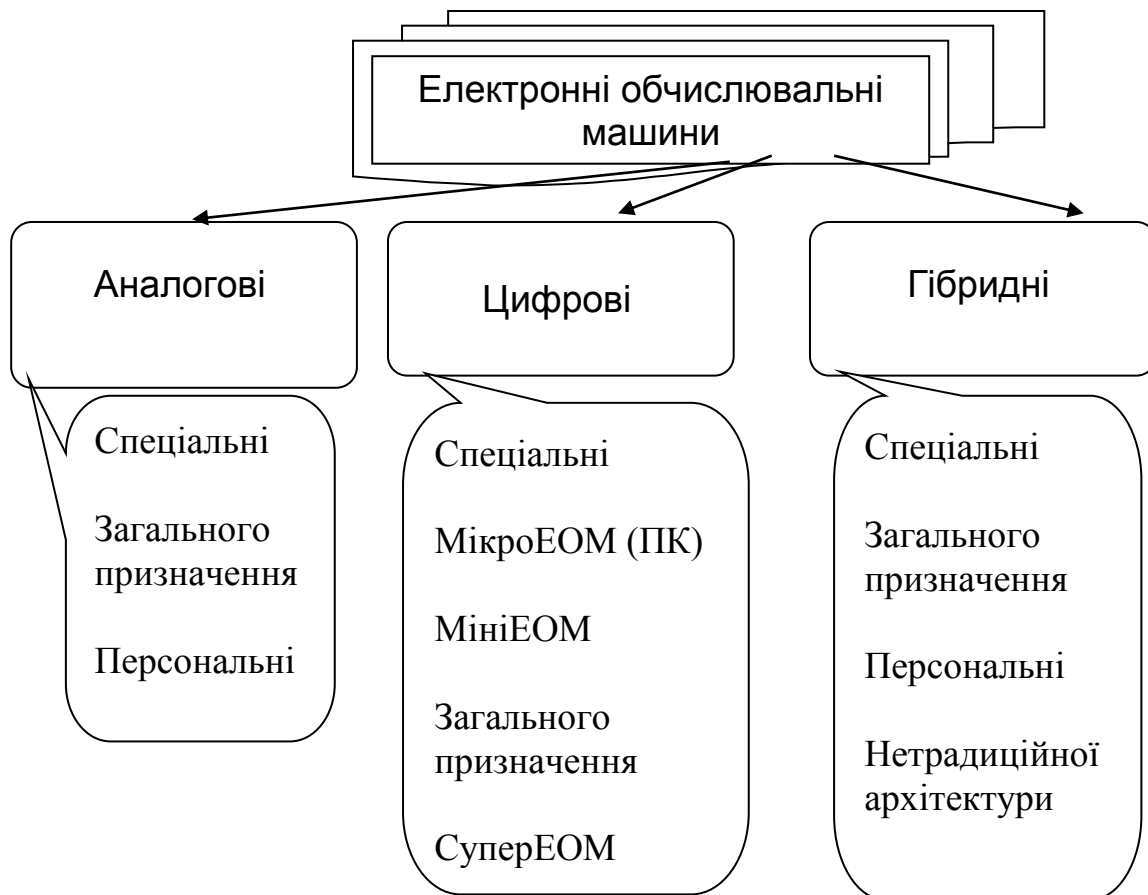
Розглянута вище класифікація є в багатьох працях, і хоча вона й відображає головну взаємодію потоків даних і команд, однак все ж не охолює всіх можливих взаємодій, які трапляються на практиці.

Наприклад, проаналізуємо конвеєрно-векторне опрацювання. Щодо потоку даних та інтегрального опрацювання дії – це тип ОКБД. Проте опрацьовувальну дію можуть виконувати підконвеєри, тобто множина послідовно виконуваних команд над кожною парою операндів, а це тип БКОД.

Можна, звичайно, ввести проміжні групи нижчого рангу й отримати ранговано однозначно визначену систему архітектур. Однак життя, як звичайно, не вписується навіть у найретельніше відпрацьовані схеми. Конструктори ОС прагнуть досягти не найгармонійнішої одноманітної архітектури, а максимальної продуктивності в заданому класі задач чи найліпшого співвідношення продуктивності ОС до її вартості, застосовуючи всі можливі архітектурні й апаратні хитрощі [1].

2.2. Класифікація за функціональним призначенням

Як уже зазначено, сучасні обчислювальні системи умовно поділяють на аналогові, дискретні та гібридні. Кожна з цих груп має ще свій внутрішній поділ, який виглядає приблизно так [2]:



2.3. Класифікація за способом організації виконання команд

CISC (Complete Instruction Set Computer) — ЕОМ з повним набором інструкцій (команд), які виконує мікропроцесор. До цього типу належить сімейство ПЕОМ на базі мікропроцесорів **x86**. Внаслідок широкого набору команд архітектура таких процесорів страждає від надміру апаратних ресурсів, склад і призначення регістрів досить неоднорідні. На виконання команд необхідно затрачати значний час, причому час виконання різних команд не однаковий (різна кількість машинних тактів). Ці фактори негативно впливають на продуктивність ПЕОМ, тому починаючи з МП **Intel486** застосовують комбіновану архітектуру, а саме у **CISC**-процесор

вкраплюють **RISC**-ядро.

Мікропроцесори з архітектурою **RISC (Reduced Instruction Set Computer)** використовують порівняно невеликий (скорочений) набір найуживаніших команд. Цей набір визначено внаслідок статистичного аналізу значної кількості програм для головних галузей застосування **CISC**-процесорів. Особливістю **RISC**-архітектури є те, що всі команди мають однаковий формат, працюють з операндами, які розташовані у регістрах процесора. Звертання до пам'яті виконується за допомогою спеціальних команд завантаження регістра і запису.

Невеликий набір команд і простота їхньої структури дають змогу реалізувати повністю апаратне виконання й ефективний конвеєр з порівняно незначним об'ємом обладнання. Завдяки цьому команди виконуються в 2-4 рази швидше, ніж у звичайних **CISC** з тою ж тактовою частотою. Продуктивність такого процесора на 30% більша ніж звичайного. Перші експериментальні МП (**RISC II, MIPS, IBM 801**) з'явилися ще у 1980-81 рр. Мікропроцесорну революцію продовжили провідні фірми у 1985-86 рр. (**Acorn, AMD, IBM, Sun Microsystems**). Сьогодні **RISC**-процесори здебільшого використовують для побудови співпроцесорів та спецпроцесорів, інтелектуальних контролерів, багатопроцесорних систем та ін. До сімейства **RISC**-процесорів належить також архітектура **MIPS** (англ. *Microprocessor without Interlocked Pipeline Stages* — мікропроцесор без блокувань в конвеєрі), яку розробляє компанія **MIPS Technologies**. Архітектуру **MIPS** використовують у комп'ютерах **SGI**, в ігрових консолях **Nintendo 64**, **Sony PlayStation**, **Sony PlayStation 2** і **Sony PSP** та вмонтованих системах.

Головна ідея **MIPS**-архітектури полягає у використанні простих команд (не передбачається апаратна підтримка операцій множення і ділення), що забезпечує високу швидкість їх виконання (висока тактова частота). Це значно спрощує будову процесора і дає змогу уникати блокувань конвеєра. На сьогодні **MIPS** використовують у вмонтовуваних пристроях з критичною продуктивністю.

В деяких серверних процесорах та графічних процесорах (англ. GPU -Graphical Processor Unit) використовують архітектуру **VLIW** (англ. very long instruction word — «дуже довга машинна команда») — архітектуру процесорів з декількома обчислювальними пристроями. Одна інструкція **VLIW**-процесора містить декілька операцій, які повинні виконуватись паралельно. В суперскалярних процесорах також є декілька обчислювальних модулів, але завдання

розподілу між ними роботи розв'язується апаратно. Це дуже ускладнює дизайн процесора і може призвести до помилок. У VLIW-процесорах завдання розподілу виконується під час компіляції і в інструкціях явно вказано, який обчислювальний пристрій має виконувати яку команду.

VLIW можна вважати логічним продовженням ідеології RISC, яке поширює її на архітектури з декількома обчислювальними модулями.

Як і в RISC, в інструкції явно вказується, що саме має робити кожний модуль процесора. Через це довжина інструкції може досягати 128 або ж 256 біт.

Розглянемо приклад VLIW-процесора з двома арифметико-логічними пристроями (АЛПр). Додамо чотири числа, які знаходяться в регістрах R1, R2, R3 і R4. Тоді псевдокод може виглядати так:

$R5=R1+R2$, $R6=R3+R4$; кожний АЛПр додає свою пару чисел
 $R0=R5+R6$, NOP ; перший АЛПр додає проміжні суми, другий простоює.

З цього прикладу можна зробити такі висновки:

1 (переваги) Підхід VLIW значно спрощує архітектуру процесора за рахунок розподілу обчислювальних пристроїв компілятором.

Оскільки з процесора видалено великі і складні вузли, які виконували розподіл обчислювальних пристроїв, значно зменшується енергоспоживання;

2 (недоліки) Через велику кількість порожніх інструкцій для невикористовуваних пристроїв програми для VLIW-процесорів можуть бути значно довшими, ніж аналогічні програми для традиційних архітектур.

Реалізації VLIW-процесорів:

1. Мікропроцесор Intel Itanium має 64-бітну систему команд «з явним паралелізмом» (англ. explicitly parallel instruction computing, EPIC), яка є одним із варіантів VLIW;

2. Процесори фірми Tilera також мають VLIW-архітектуру;

3. Графічні процесори AMD/ATI Radeon, починаючи з R600, мають VLIW-архітектуру.

Ще один тип ЕОМ — з **MISC-архітектурою (Minimise Instruction Set Computer)**, які мають мінімальний набір інструкцій (Сетунь, Forth).

Список літератури до теми

1. *Смирнов А.Д.*, Архитектура вычислительных систем. Учеб. пособие для вузов. – М., 1990.
2. *Аладьев В.З., Хунт Ю.Я., Шишаков М.Л.* Основы информатики.– Учеб. пособие.– М., 1999.
3. *Каган Б.М.* Электронные вычислительные машины и системы: Учеб.пособие для вузов. – М., 1991.
4. *Рикалюк Р.Є., Стягар О.М., Данчак П.В.* Вступ до комп'ютерних мереж: Текст лекцій. – Львів, 1996.
5. *Буров Є.* Комп'ютерні мережі. – Львів, 1999.
6. *Колодницький М.М.* Технічне та програмне забезпечення комп'ютерних інформаційних технологій. – Житомир, 1995.

Контрольні запитання до теми

1. У чому полягає суть класифікації за інтегральними ознаками?
2. Яка залежність продуктивності ЕОМ від кількості процесорів?
3. Наведіть приклади мас-процесорних систем.
4. Які типи взаємодії потоку команд і потоку даних?
5. Як класифікують обчислювальні системи за функціональним призначенням?



IBM Blue Gene P supercomputer [CC BY-SA 2.0](#) view terms

[Blue Gene / P](#) From [Argonne National Laboratory](#) Uploaded using [F](#)

Applications of supercomputers

The stages of supercomputer application may be summarized in the following table:

Decade	Uses and computer involved
1970s	Weather forecasting, aerodynamic research (Cray-1). ^[75]
1980s	Probabilistic analysis, ^[76] radiation shielding modeling ^[77] (CDC Cyber).
1990s	Brute force code breaking (EFF DES cracker). ^[78]
2000s	3D nuclear test simulations as a substitute for legal conduct Nuclear Non-Proliferation Treaty (ASCI Q). ^[79]
2010s	Molecular Dynamics Simulation (Tianhe-1A). ^[80]