

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КІРОВОГРАДСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Паралельні та розподілені обчислення

Курс лекцій

з елементами кредитно – модульної
системи організації навчального процесу

*для студентів денної та заочної форми навчання
за напрямом підготовки 6.050102 «Комп'ютерна інженерія»*

Укладач:
Доцент

Смірнова Н.В.

Кіровоград 2015

ВСТУП

Для розв'язання багатьох задач (прогноз погоди, задачі гідро- і газодинаміки, квантової хімії, астрономії, спектроскопії, біології, ядерної фізики) необхідна висока продуктивність та висока швидкість передачі інформації по каналах зв'язку, великі об'єми оперативної і постійної пам'яті, які не можуть забезпечити типові обчислювальні засоби. Одним з шляхів забезпечення таких вимог є організація паралельних та розподілених обчислень і відповідних технічних засобів їх реалізації.

Причому, ефективність паралельної обробки залежить як від продуктивності комп'ютерів, так і від розмірів і структури пам'яті, пропускної здатності каналів зв'язку, використаних мов програмування, компіляторів, операційних систем, чисельних методів та інших математичних досліджень. Такий широкий обсяг параметрів вимагає проведення досліджень на різних рівнях: на рівні розпаралелення алгоритмів, створення спеціальних мов програмування, компіляторів, багатопроцесорних систем, неоднорідних систем, кластерів і систем, що розподілені на великих територіях.

Метою вивчення дисципліни є засвоєння основних методів та алгоритмів організації паралельних та розподілених обчислень, принципів побудови відповідних структур, набуття початкових практичних навиків проектування таких засобів.

В **результаті вивчення** курсу студент повинен знати основні методи, алгоритми і засоби паралельної та розподіленої обробки інформації, засоби програмування на паралельних та розподілених структурах, склад апаратних засобів та програмного забезпечення обчислювальних систем з елементами паралельної та розподіленої обробки і класи мов програмування високого рівня для них; вміти виконувати елементарні вправи з розпаралелення задач та алгоритмів, проводити розрахунки параметрів процесорів, проектувати окремі вузли.

В розділі курсу "**Організація паралельних обчислень**" розглядаються такі основні питання:

- основні поняття про паралельні обчислення;
- методи оцінки продуктивності паралельних алгоритмів;
- розробка паралельного алгоритму;
- структури зв'язку між процесорами;
- основні класи паралельних комп'ютерів;
- схеми паралельних алгоритмів задач;
- мови паралельного програмування.

Тема №1: Основні поняття про паралельні обчислення**Питання:**

- 1. Поняття про паралельні та розподілені обчислення**
- 2. Області застосування і задачі паралельної обробки**
- 3. Моделі паралельних обчислень. Конвексризація і паралелізм**
- 4. Засоби для проведення паралельних обчислень**
- 5. Рівні розпаралелення**
- 6. Паралельні операції**
- 7. Основні принципи паралелізму (розпаралелення)**
- 8. Класифікація структур паралельної обробки**

Вправи і завдання до теми №1**1. Поняття про паралельні та розподілені обчислення**

В залежності від предметної області застосування є багато визначень термінів, які характеризують паралельні та розподілені обчислення. На основі аналізу літературних джерел і варіантів практичної реалізації можна так визначити ці терміни:

Паралельні обчислення – обчислення, що підтримуються на математичному, алгоритмічному, програмному чи апаратному рівні (на всіх або декількох) і забезпечують можливість паралельного виконання задачі.

В [1] під терміном “паралельні обчислення” розуміється сукупність питань, які відносяться до створення ресурсів паралелізму в процесах розв’язання задач і гнучкому керуванню реалізацій цього паралелізму з метою досягнення найбільшої ефективності обчислювальної техніки.

Розподілені обчислення – обчислення, які підтримуються стандартними чи закритими протоколами обміну та незалежними апаратними засобами (комп’ютери, сервери), що представляються користувачу єдиним обчислювачем, придатним для вирішення складної задачі.

Стосовно використаних ресурсів можна стверджувати: здебільшого паралельні структури реалізуються на спеціалізованих процесорах, розподілені структури – на універсальних (стандартних) комп’ютерах (серверах), які об’єднані в мережі різного типу.

2. Області застосування і задачі паралельної обробки

Є коло обчислювальних задач, для розв’язку яких необхідні потужніші обчислювальні ресурси, ніж ті, які можуть забезпечити типові комп’ютери чи системи. В таких задачах необхідно забезпечити: *надвисоку швидкодію, великий об’єм оперативної пам’яті, велику кількість інформації, що передається, обробку і зберігання великого об’єму інформації*. При наявності хоча б однієї з наведених вимог використання паралельної обробки оправдано.

Приклади:

1. Складні, багатовимірні задачі, які необхідно розв’язати на протязі досить обмеженого часу, вимагають забезпечення *надвисокої швидкодії*, наприклад - задачі прогнозу погоди. Область розв’язку (атмосфера) розбивається на окремі просторові зони. Причому, для забезпечення прогнозу на певному періоді часу, обчислення в кожній зоні повторюється багато разів. Якщо об’єм зони рівний 1 км^3 , то для моделювання 10 км атмосфери необхідно 5×10^8 таких зон. Припустимо, що обчислення в кожній зоні вимагає 200 операцій з рухомою крапкою, тоді за один часовий крок необхідно виконати 10^{11} операцій з рухомою крапкою. Для того, щоб зпрогнозувати погоду з передбачуваністю 10 днів з 10-ти хвилинним кроком комп’ютеру з продуктивністю 100 Mflops (10^8 операцій з рухомою крапкою за секунду) необхідно 10^7 секунд чи понад 100 днів. Для того, щоб провести розрахунок за 10 хв., необхідний комп’ютер продуктивністю 1.7 Tflops (1.7×10^{12} операцій з рухомою крапкою за секунду).

2. До категорії задач, що вимагають *великого об'єму оперативної пам'яті*, відносяться, наприклад, задачі гідро- і газодинаміки з розрахунку течій з врахуванням різних фізичних і хімічних процесів. Такі задачі є, як правило, багатовимірними, і розрахунок по кожному з напрямків вимагає оперативної пам'яті понад 10 Гбайт. В квантовій хімії неемпіричні розрахунки електронної структури молекул вимагають обчислювальних затрат, пропорційних N^4 чи N^5 , де N умовно характеризує кількість молекул.

3. Вимога *забезпечення великої кількості інформації, що передається* характерна для задач гідро- і газодинаміки з змінюючими граничними умовами, коли обчислювальний алгоритм постійно вимагає підведення нової інформації, і задач економічної оптимізації, що описують поведінку системи, яка занурена в середовище з неперервно змінюючими властивостями, від яких залежить стан системи.

4. Проблема *обробки і зберігання великого об'єму інформації* характерна для задач астрономії, спектроскопії, біології, ядерної фізики.

3. Моделі паралельних обчислень. Конвеєризація і паралелізм

Конвеєризація – метод, що забезпечує сукупність різних дій за рахунок їх розбиття на підфункції з зміщенням в часі виконанням. Конвеєрний пристрій розробляють з окремих ступенів, час спрацювання яких в ідеальному випадку повинен бути однаковим.

Паралелізм – метод, що забезпечує виконання різних функцій шляхом їх розбиття на підфункції з одночасним виконанням в часі.

Приклад:

Варіант структурної схем виконання виразу $y = \sum_{i=1}^N a_i * b_i$ при паралельній обробці наведений на рис.1.1.

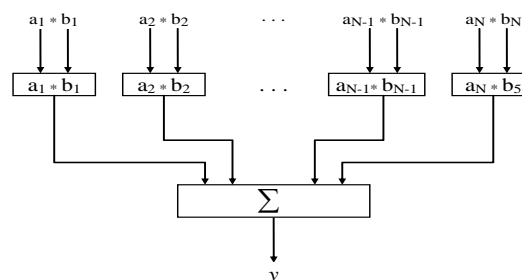


Рис. 1.1

4. Засоби для проведення паралельних обчислень

1. Апаратні засоби:

- засоби для проведення обчислень (обчислювальна техніка):
 - обчислювальна техніка, зібрана з стандартних комплектуючих;
 - обчислювальна техніка, зібрана з спеціальних комплектуючих;
- засоби візуалізації;
- засоби для зберігання і обробки даних.

2. Програмні засоби:

- програмні засоби загального призначення (операційні системи: стандартні бібліотеки, мови програмування, компілятори, профайлери, відлагоджувачі і т.п.);
- спеціальні програмні засоби: бібліотеки ([PVM](#), [MPI](#)); засоби об'єднання ресурсів ([Dynamite](#), [Globus](#) і ін.)

5. Рівні розпаралелювання

Класифікація паралельності за рівнями, що відрізняються показниками абстрактності розпаралелення задач наведена в табл.1.1. Чим "нижчий" рівень паралельності, тим детальнішим, малоелементнішим буде розпаралелення, що торкається елементів програми (інструкція, елементи інструкції тощо).

Таблиця 1.1 Рівні паралельності

Зернистість	Рівні	Об'єкт обробки	Приклад системи
Великоблокова	Програмний	Робота/Задача	Мультизадачна ОС
	Процедурний	Процес	MIMD-система
Дрібноблокова	Рівень формул	Інструкція	SIMD-система
	Біт-рівень	В межах інструкції	Машина фон Ноймана

Методи і конструктиви даного рівня обмежуються тільки цим рівнем і не можуть бути поширені на інші рівні.

Програмний рівень

На цьому рівні одночасно (або щонайменше з часовим розподілом) виконуються комплексні програми (рис.1.2). Комп'ютер, що виконує ці програми *не обов'язково повинен мати паралельну структуру*, достатньо, щоб на ньому була встановлена багатозадачна операційна система (наприклад, реалізована як система *розподілу часу*). В цій системі кожному користувачеві, відповідно до його пріоритету, планувальник (*Scheduler*) виділяє відрізок часу різної тривалості. Користувач одержує ресурси центрального процесорного блоку тільки впродовж короткого часу, а потім стає в чергу на обслуговування.

У тому випадку, коли в системі недостатня кількість процесорів для всіх користувачів (або процесів), що, як правило, найбільш імовірно, в системі моделюється паралельне обслуговування користувачів за допомогою "квазіпаралельних" процесів.

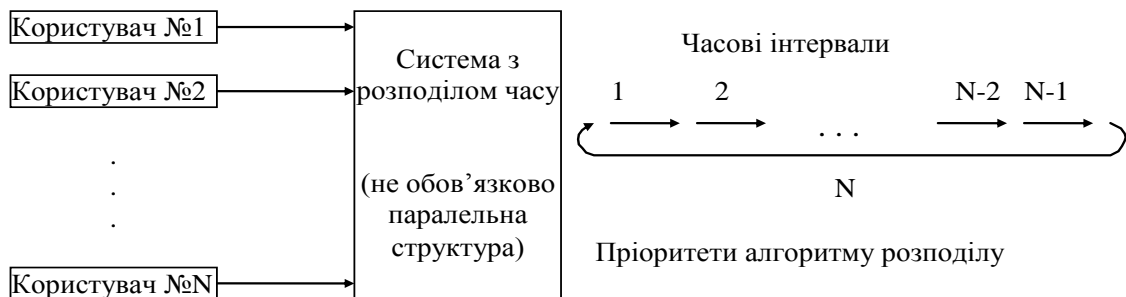


Рис.1.2. Паралельність на програмному рівні

Рівень процедур

На цьому рівні різні частини ("процеси") однієї і тієї ж програми виконуються паралельно; кількість операцій обміну даними між процесами повинна бути мінімальною.

Основне застосування процедурного рівня розпаралелення - загальне паралельне оброблення інформації, де застосовується ділення вирішуваної проблеми на паралельні задачі - частини, які вирішуються багатьма процесорами з метою підвищення обчислювальної продуктивності (приклад - рис.1.3.).

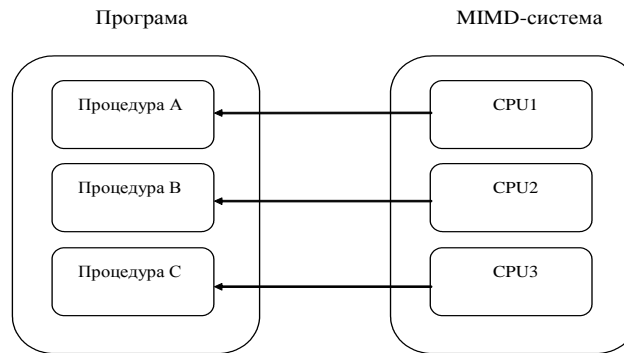


Рис.1.3. Паралельність на рівні процедур

Рівень формул (арифметичних виразів)

Арифметичні вирази виконуються паралельно покомпонентно, причому в суттєво простіших синхронних методах. Якщо, наприклад, йдеться про додавання матриць (рис.1.4.), то операція синхронно розпаралелюється просто тому, що на кожному процесорі обчислюється один (чи кілька) елемент матриці.

При застосуванні $n \times n$ процесорних елементів можна одержати суму двох матриць розмірністю $n \times n$ за час виконання однієї операції додавання (за винятком часу, потрібного на виконання операцій читання та запису даних). Цьому рівню притаманні засоби векторизації так званої *паралельності даних*. Майже кожному елементу даних тут підпорядковується свій процесор, завдяки чому ті дані, що в машині фон Ноймана були пасивними, перетворюються на “активні обчислювальні пристрої”.

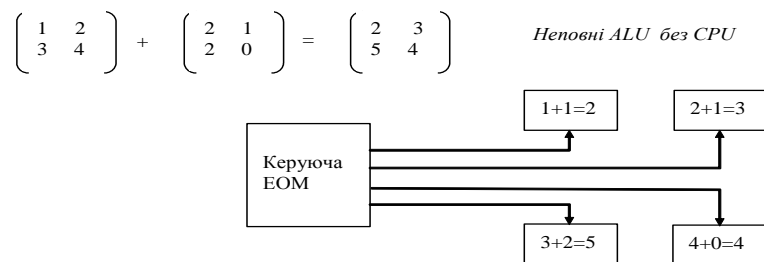


Рис.1.4. Паралельність на рівні операцій (формул)

Біт – рівень

На цьому рівні відбувається паралельне виконання бітових операцій в межах одного інформаційного слова (рис.1.5). Паралельність на рівні бітів можна знайти в будь-якому працюючому мікропроцесорі. Наприклад, у 8-розрядному арифметико-логічному пристрої побітова обробка виконується паралельними апаратними засобами.

$$\begin{array}{r} \text{AND} \quad 01011101 \\ \quad \quad 11011000 \\ \hline \quad \quad 01011000 \end{array}$$

Рис.1.5. Паралельність на рівні бітів

6. Паралельні операції

Інший вид паралельності виникає з аналізу математичних операцій над окремими елементами даних або над групами даних. Розрізняють *скалярні дані*, операції над якими виконуються послідовно, і *векторні дані*, операції над якими виконуються паралельно.

Прості операції над векторами, наприклад додавання двох векторів, можуть виконуватись синхронно і паралельно. У цьому випадку кожному елементу вектора підпорядковується один процесор. При складніших операціях, таких як формування часткових сум, побудова

ефективного паралельного алгоритму є складнішою. Розрізняють одномісцеві (монадні) та двомісцеві (діадні) операції, характерні приклади для яких наведено нижче.

Одномісцеві операції

а). Скаляр \rightarrow скаляр

Приклад $9 \rightarrow 3$

Послідовне виконання

"Корінь"

б). Скаляр \rightarrow вектор

Приклад $9 \rightarrow (9,9,9,9)$

Розмноження числової величини

"Broadcast"

в). Вектор \rightarrow скаляр

Приклад $(1,2,3,4) \rightarrow 10$

Редукція вектора в скаляр

"Складання"

(із припущенням, що довжина вектора не змінюється)

г-1) Локальна векторна покомпонентна операція.

Приклад: $(1,4,9,16) \rightarrow (1,2,3,4)$

"Корінь"

г-2) Глобальна векторна операція з перестановками.

Приклад: $(1,2,3,4) \rightarrow (2,4,3,1)$

"Заміна місць компонент вектора"

г-3) Глобальна векторна операція (часто складається з простих операцій)

Приклад: $(1,2,3,4) \rightarrow (1,3,6,10)$

"Часткові суми"

Двомісцеві операції

д) (скаляр, скаляр) \rightarrow вектор

Приклад: $(1,2) \rightarrow 3$

Послідовне виконання

"Скалярне додавання"

е) (скаляр, вектор) \rightarrow вектор

Приклад: $(3, (1,2,3,4)) \rightarrow (4,5,6,7)$

Покомпонентне застосування операцій над скаляром і вектором

"Додавання скаляра з вектором"

Операція виконується як послідовність операцій б) та є)

додавання векторів:

$(3,3,3,3)$

+

$(1,2,3,4)$

$(4,5,6,7)$

є) (вектор, вектор) \rightarrow вектор

Покомпонентне застосування операції над двома векторами

Приклад: $((1,2,3,4), (0,1,3,2)) \rightarrow (1,3,6,6)$

"Додавання векторів"

7. Основні принципи паралелізму (розпаралелення)

Розпаралелити кожну задачу можна не єдиним способом. Тому доцільно розглянути алгоритми можливого розпаралелення типових задач незалежно від конкретної програмної і платформенної реалізації. В загальному випадку процедура розпаралелення розбі'ється на 3 етапи:

- розбиття задачі на незалежні під задачі;
- призначення конкретних процесорів для виконання кожної під задачі;
- збирання результатів роботи окремих процесорів.

8. Класифікація структур паралельної обробки

Є різні системи класифікації паралельних систем, які базуються на різнотипних головних ознаках до класифікації. Багато з них базується на класифікації М. Флінна (1966 р.), де паралельна обробка виконується на SIMD і MIMD архітектурах.

В сучасних класифікаційних системах архітектури, які попадають в один клас, відрізняються за кількістю процесорів, природі і топології зв'язку між ними, за способом організації пам'яті, за технологією програмування та іншими ознаками.

Наприклад, А.Базу (A.Basu) вважає, що будь-яку паралельну обчислювальну систему можна однозначно описати послідовністю рішень, прийнятих на етапі її проектування, а сам процес проектування представити у виді дерева. Класифікація за Базу наведена на рис.1.6.

Паралельні обчислювальні системи

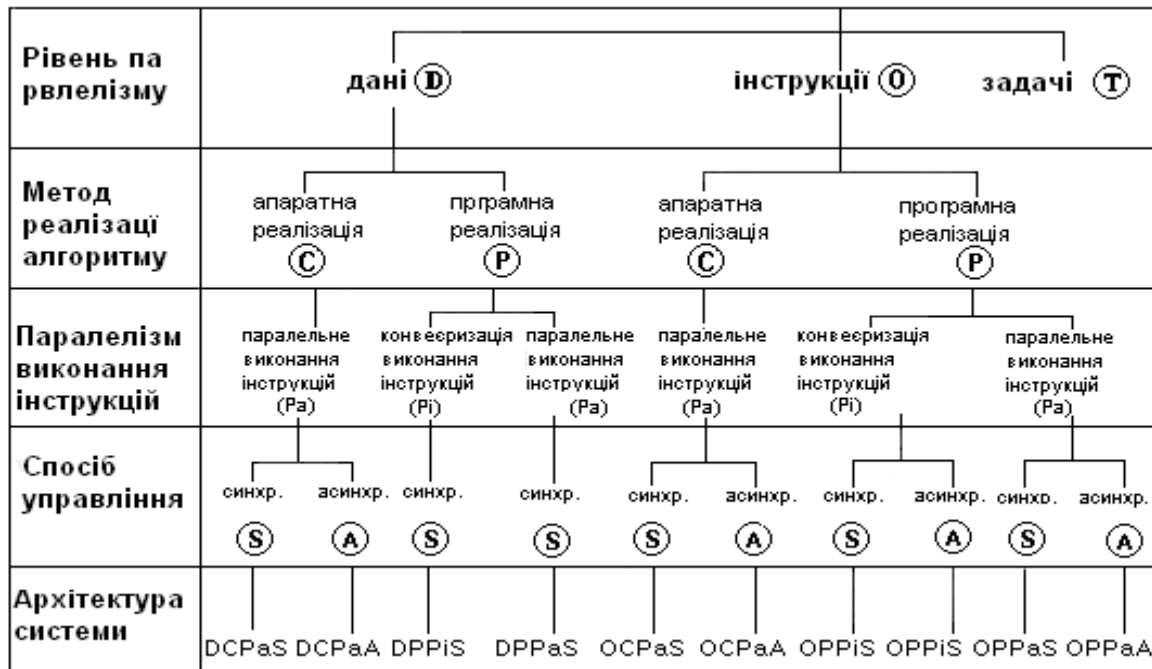


Рис.1.6. Класифікація за Базу

Р.Дункан визначає такий набір вимог на який може спиратися класифікація (див.рис.1.7):

З класу паралельних машин повинні бути виключені ті, у яких паралелізм закладений лише на найнижчому рівні, включаючи:

- конвеєризацію на етапі підготовки і виконання команди (instruction pipelining), тобто часткове перекриття таких етапів, як дешифрація команди, обчислення адрес операндів, вибірка операндів, виконання команди і збереження результату;
- наявність в архітектурі декількох функціональних пристроїв, що працюють незалежно, зокрема, можливість паралельного виконання логічних і арифметичних операцій;
- наявність окремих процесорів вводу/виводу, що працюють незалежно і паралельно з основними процесорами.

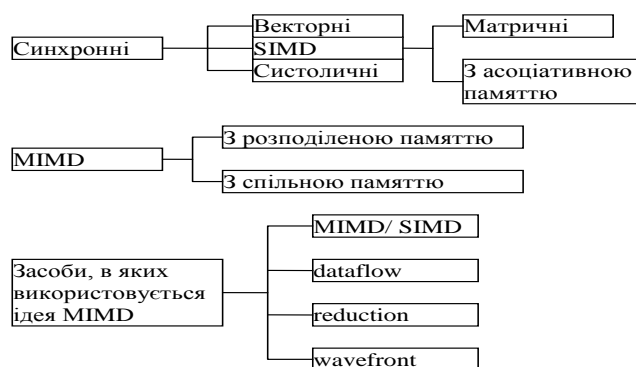


Рис.1.7 Вимоги до класифікації за Р. Дунканом

Дункан дає неформальне визначення паралельної архітектури, а саме: *паралельна архітектура* - це такий спосіб організації обчислювальної системи, при якому допускається, щоб безліч процесорів (простих або складних) може працювати одночасно, взаємодіючи в міру потреби один з одним.

Розглянемо три види архітектур, в яких використовують нетрадиційні моделі обчислень.

Dataflow - використовують модель, у якій команда може виконуватися відразу ж, як тільки обчислені необхідні операнди. Таким чином, послідовність виконання команд визначається залежністю за даними, що може бути виражена, наприклад, у формі графа.

Модель обчислень, застосовувана в *reduction* машинах полягає в наступному: команда стає доступною для виконання тоді і тільки тоді, коли результат її роботи потрібно іншій, доступній для виконання, команді як операнд.

Wavefront array архітектура поєднує в собі ідею систолічної обробки даних і модель обчислень, що використовується в *dataflow*. У даній архітектурі процесори об'єднуються в модулі і фіксуються зв'язки, по яких процесори можуть взаємодіяти один з одним. Однак, на противагу ритмічній роботі систолічних масивів, дана архітектура використовує асинхронний механізм зв'язку з підтвердженням (handshaking), через що "фронт хвилі" обчислень може змінювати свою форму в міру проходження по всіх процесорах.

Е.Кришнамарфі для класифікації паралельних обчислювальних систем пропонує використовувати чотири характеристики, які подібні до характеристик класифікації А.Бази: *ступінь гранулярності, спосіб реалізації паралелізму, топологія і природа зв'язку, спосіб керування процесорами*.

На основі виділених чотирьох характеристик можна визначити місце найбільш відомих класів архітектур у даній класифікації (див. табл.1.2).

Таблиця 1.2

Тип архітектури	Гранулярність	Реалізація паралелізму	Зв'язок процесорів	Спосіб керування
Векторно-конвеєрні комп'ютери	На рівні даних	Апаратна	Проста топологія із середньою зв'язністю	Синхронний
Класичні мультипроцесори	На рівні задач	Комбінована	Проста топологія зі слабкою зв'язністю	Асинхронний
Матриці процесорів	На рівні даних	Апаратна	Двовимірні масиви із сильною зв'язністю	Синхронний
Систолічні масиви	На рівні даних	Апаратна	Складна топологія із сильною зв'язністю	

Незважаючи на те, що класифікація Е. Кришнамарфі побудована лише на чотирьох ознаках, вона дозволяє виділити і описати такі "нетрадиційні" паралельні системи, як систолічні масиви, машини типу *dataflow* і *wavefront*.

Заслужують на увагу класифікації *Хандлера (ерлангівська схема)* [8], *Р. Хокні* [1].

Вправи і завдання до теми №1

1. Як співвідносяться між собою класифікації *Хендлера* і *Флінна*?
2. Які параметри архітектури паралельного комп'ютера треба знати користувачу для створення ефективних програм? Запропонуйте класифікацію комп'ютерів, яка спирається на виділені параметри.
3. Чи можна організувати паралельні обчислення для розв'язання задачі знаходження найменшого числа з трьох чисел?, з чотирьох чисел? Якщо можна, то яким чином?
4. Наведіть приклад застосування паралельного опрацювання для будь-якої галузі науки і техніки. Які з вимог паралельної обробки там використовуються?
5. Чи є принтер засобом для організації паралельної обробки?
6. Необхідно перемножити матрицю **A** ($n_1 \times n_2$) на матрицю **B** ($n_2 \times n_3$). Розробіть функціональну схему пристрою, який забезпечить найшвидше виконання даної операції.
7. Чому в конвеєрних пристроях тривалість спрацювань окремих ступенів роблять однаковими?
8. Один робітник може вирити яму розміром $1 \times 1 \times 1 \text{ м}^3$ за годину. За скільки часу бригада з 5, 10, 20 робітників вириють яму розміром $2 \times 2 \text{ м}^2$ і глибиною 1м, якщо продуктивність всіх робітників однакова?
9. Побудуйте графік залежності виконання роботи від кількості робітників в бригаді.