

Практична робота 4. Архітектура паралельних комп'ютерних систем

Мета: Аналіз паралельних обчислювальних систем.

Теоретичні відомості

В основі паралельних обчислювальних систем (ОС) полягає ідея, що вирішення задачі на паралельних процесорах забере в p - разів менше часу, ніж на одному комп'ютері. Ця ідея довгий час не могла бути реалізована у повному обсязі через відсутність потужної елементної та конструкторсько-технологічної бази, а також відповідного системного та прикладного програмного забезпечення. Положення різко змінилось зі створенням великих та надвеликих інтегральних схем, мікропроцесорів, мікро-ЕОМ, а також розвитку паралельного програмування та паралельних обчислювальних методів.

Для побудови паралельних ОС використовують серійні мікропроцесори та мікро-ЕОМ, а також розробляють відповідним чином орієнтовані мікропроцесори та мікро-ЕОМ (трансп'ютери). Позитивною якістю паралельних ОС є їх нарощуваність.

Важливою класифікаційною ознакою паралельних ОС є спосіб керування сукупністю процесорів. В обчислювальних системах типу ОКМД (одиначний потік команд – множинний потік даних) всі p процесорів ОС знаходяться під керуванням головного, керуючого процесора. У кожному такті всі процесори виконують одну й ту саму команду або простоюють. Таким чином, в одному потоці команд обробляється багато потоків даних, що проходять через процесори. До них відносять також ОС, процесори яких виконують однакові програми й обмінюються між собою даними синхронно.

У паралельних ОС класу МКМД (множинний потік команд – множинний потік даних) окремі процесори працюють під керуванням своїх власних пристроїв керування і виконують різні гілки програм.

Іншою важливою ознакою архітектури паралельних ОС є тип пам'яті – загальна або локальна. У паралельних ОС із загальною пам'яттю всі процесорні елементи (ПЕ) мають пам'ять, що ПЕ ділять між собою для доступу (рис. 4.1).

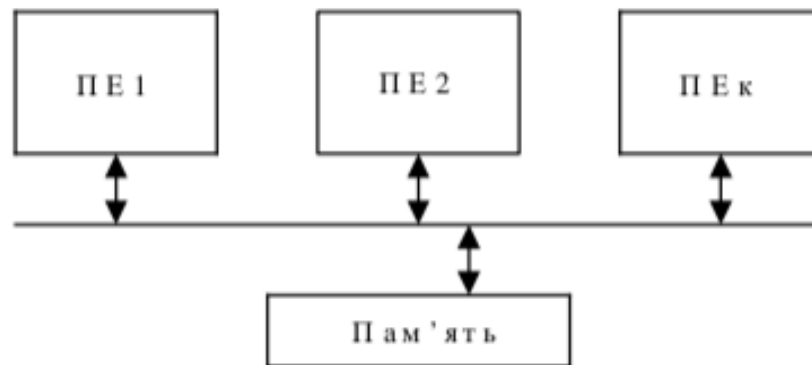


Рисунок 4.1 – Структура паралельної ОС із загальною пам'яттю

Перевагою таких ОС є швидкий обмін даними між процесорами. Недоліком є затримки доступу до пам'яті при одночасному звертанні різних процесорів, при цьому тривалість затримки збільшується з ростом кількості процесорів. Цій недолік можна мінімізувати, якщо кожному процесору надати кеш-пам'ять (рис. 4.2.).

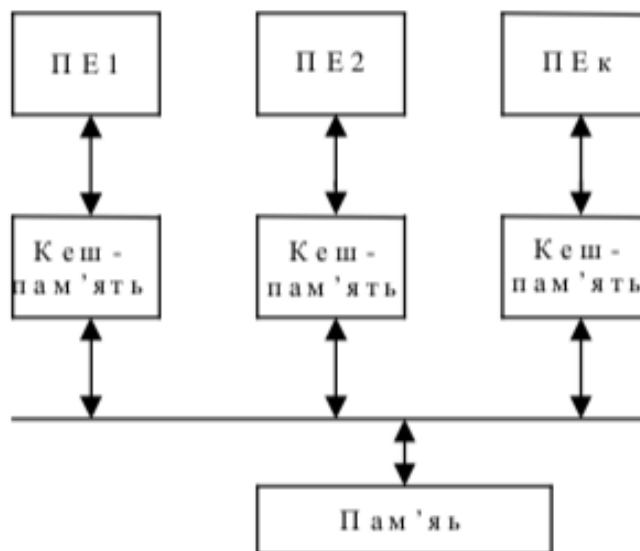


Рисунок 4.2 – Структура паралельної ОС із кеш-пам'яттю

Кожний ПЕ може мати власну локальну пам'ять для збереження своїх програм і проміжних результатів. Тоді загальна пам'ять використовується для

даних, необхідних більш ніж одному ПЕ. Всі взаємодії між ПЕ виконуються через загальну пам'ять. Її недолік полягає в тому, що кілька процесорів можуть вимагати доступ до цієї пам'яті одночасно. Дозвіл отримує тільки один ПЕ. У цьому випадку виникають конфліктні затримки, пов'язані із забезпеченням доступу до пам'яті, які можуть зростати зі збільшенням кількості процесорів.

Для зменшення конфліктів використовується розподілена пам'ять банків з локально-адресованими просторами (рис. 4.3).



Рисунок 4.3 – Структура паралельної ОС із розподіленою пам'яттю банків з локально-адресованими просторами

Розподілена пам'ять поділена на банки з локально-адресованими просторами. Це дозволяє з деякою затримкою у часі майже одночасно звертатись різним ПЕ до різних банків. Комутаційна мережа необхідна для підключення ПЕ до різних банків пам'яті під час виконання паралельної програми.

Як комутаційна мережа ОС із розподіленою пам'яттю може застосовуватись багатошинна топологія та перехресна комутація.

Багатошинна топологія передбачає наявність m незалежних шин для підключення до m банків пам'яті. (рис. 4.4). У цьому випадку виникають конфлікти при захваті шини декількома ПЕ, що є серйозною проблемою при

збільшенні кількості ПЕ. Це достатньо громіздка структура. Перевагою такого способу комутації є висока надійність. Така топологія цілком прийнятна для високопродуктивних ОС. Пропускна здатність зростає пропорційно кількості шин. У порівнянні з одношинною архітектурою управління мережею з декількома шинами складніше через необхідність запобігання конфліктів, що виникають, коли в парах вузлів, що обмінюються по різних шинах, присутній загальний вузол.



Рисунок 4.4 – Шинна топологія ОС

Перехресний комутатор (рис. 4.5) дозволяє домогтися повної зв'язаності ОС за рахунок міжпроцесорних обмінів через загальні банки пам'яті. Вертикальні лінії перехресного комутатора підключені до m блоків пам'яті, а горизонтальні – до n процесорів. На перетинанні вертикальних і горизонтальних ліній установлені перемикачі. Однак їх число, рівне $n*m$, для більших систем стає непрактичним.

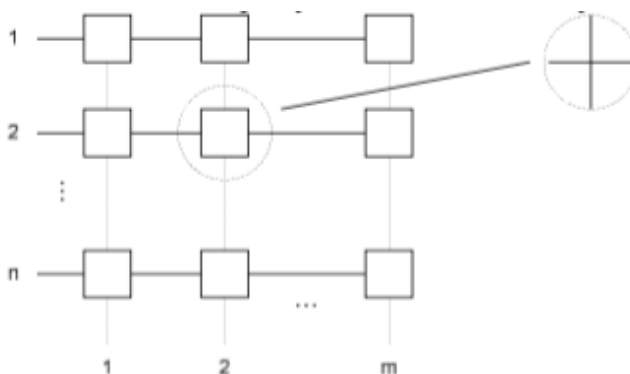


Рисунок 4.5 – Топологія перехресної комутації

Використання перехресної комутації ОС на основі матричного комутатора являє собою класичний приклад одноступінчатої динамічної мережі. Матричний комутатор використовується для зв'язку ПЕ з банками пам'яті. Це найшвидший

спосіб доступу ПЕ до банків пам'яті. Недоліками є висока вартість та залежність надійності ОС від надійності комутатора.

При використанні розподіленої пам'яті кількість ПЕ, що одночасно отримують дозвіл на доступ до пам'яті дорівнює кількості банків цієї пам'яті. Конфлікти до пам'яті зберігаються лише у випадках одночасного доступу декількох ПЕ до одного й того ж банку пам'яті. Таким чином, структура системи на рис. 4.3 є найбільш досконалою, але нарощуваність (масштабованість) цих систем обмежена (до 64 ПЕ).

Повністю конфлікти доступу до пам'яті усунуті в архітектурах із локальною пам'яттю. У таких системах кожний процесор має власну пам'ять та адресується лише до неї. Процесори обмінюються між собою шляхом передавання відповідних повідомлень. Такі системи мають необмежену масштабованість.

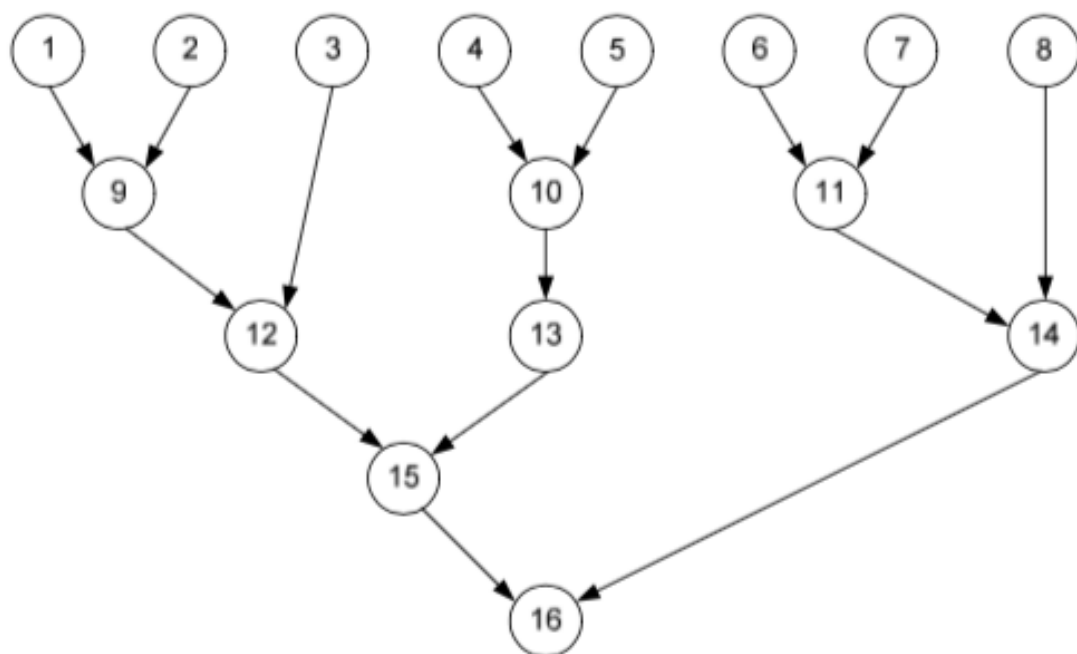
Можна виділити 4 переваги паралельних ОС у порівнянні із послідовними:

1. Паралельна архітектура – єдиний спосіб побудови надпродуктивних ОС.
2. Відношення продуктивність/вартість у паралельних ОС вищий, ніж у послідовних.
3. Паралельні ОС дозволяють нарощувати продуктивність в результаті модульної конструкції.
4. Паралельні ОС є відмовостійкими.

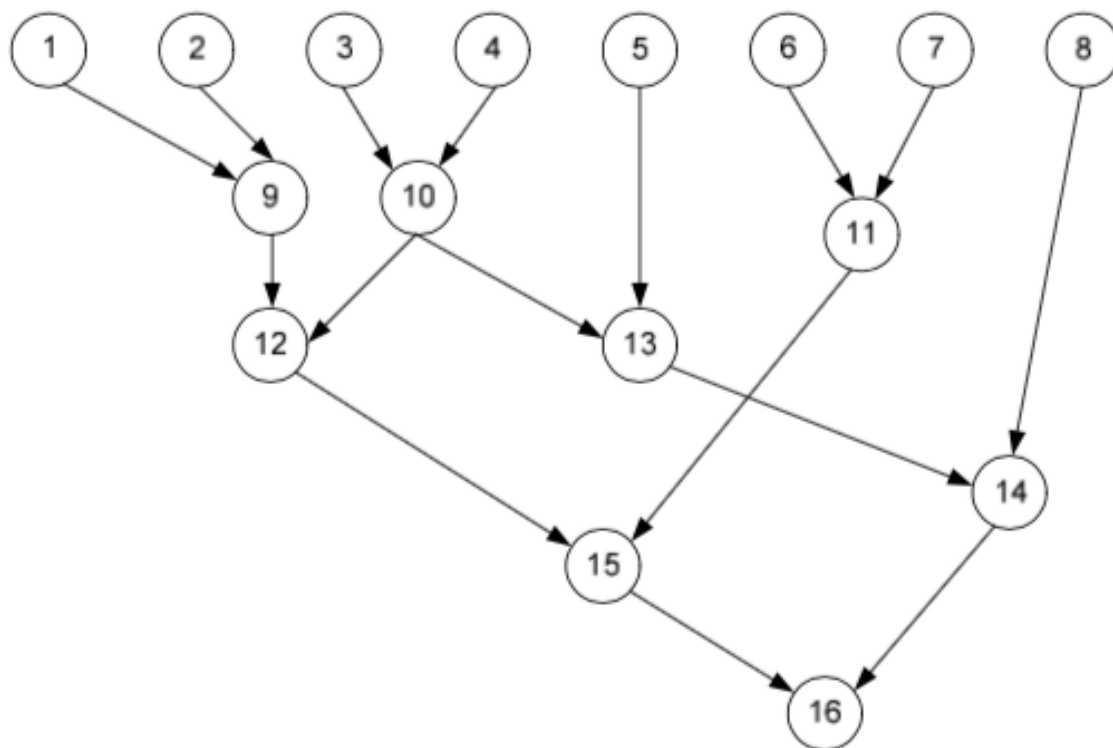
Вихідні дані:

а) граф обчислювальної задачі (рис. 4.6, 4.7).

Вихідні дані визначають за останньою цифрою залікової книжки. Остання цифра залікової книжки визначає граф обчислювальної задачі. Цифрам 0, 1 відповідає граф на рис. 4.6, цифрам 2, 3 відповідає граф на рис. 4.7, цифрам 4, 5 відповідає граф на рис. 4.8, цифрам 6, 7 відповідає граф на рис. 4.9, цифрам 8, 9 відповідає граф на рис. 4.10.

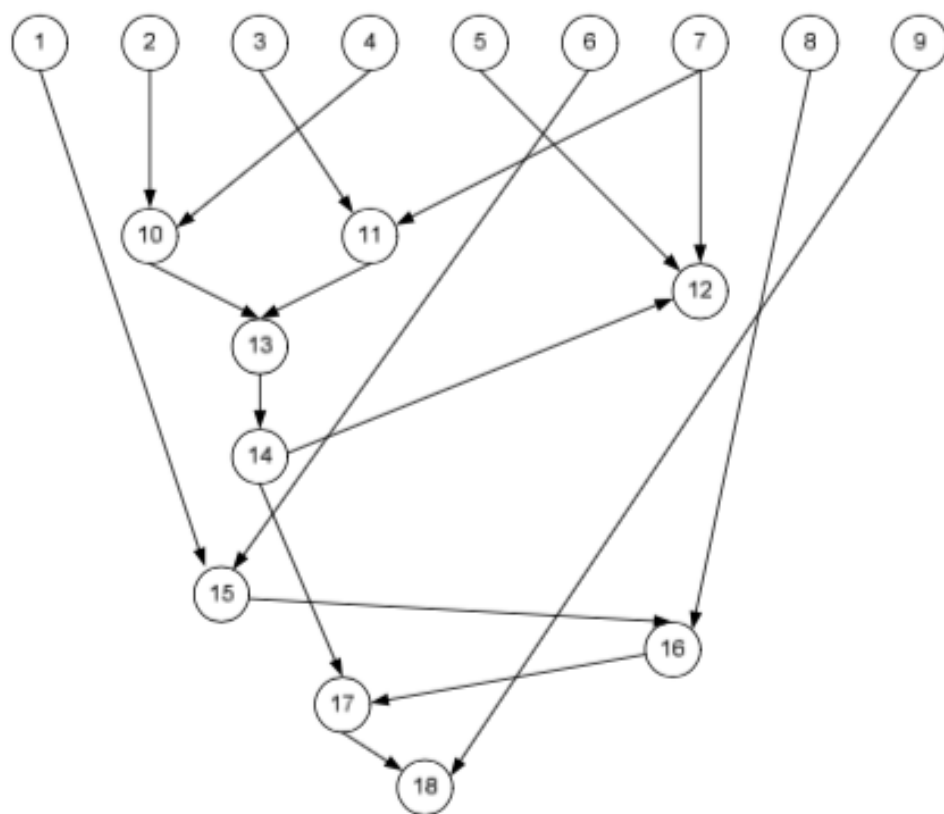


a

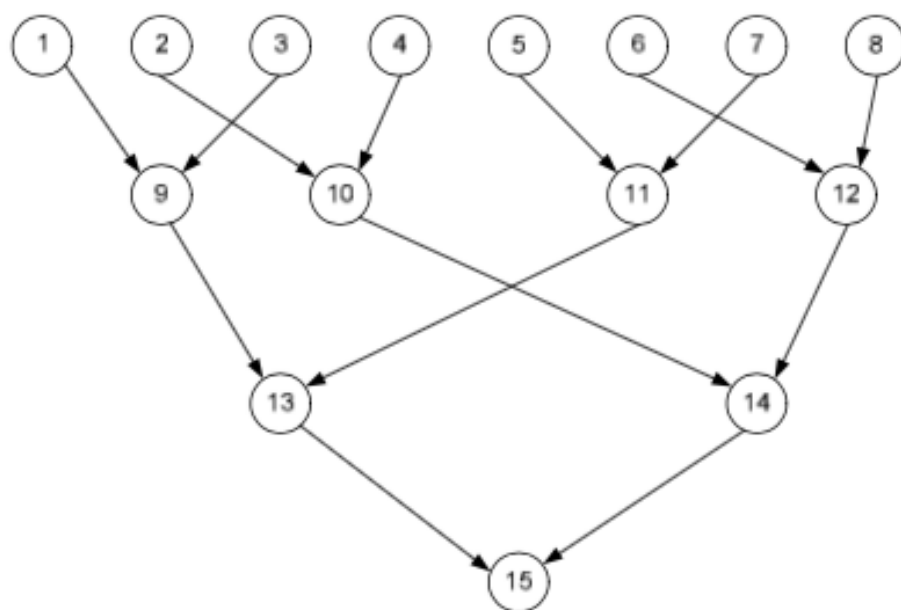


б

Рисунок 7.6 – Граф обчислювальної задачі: а – остання цифра залікової книжки 0; б – остання цифра залікової книжки 1

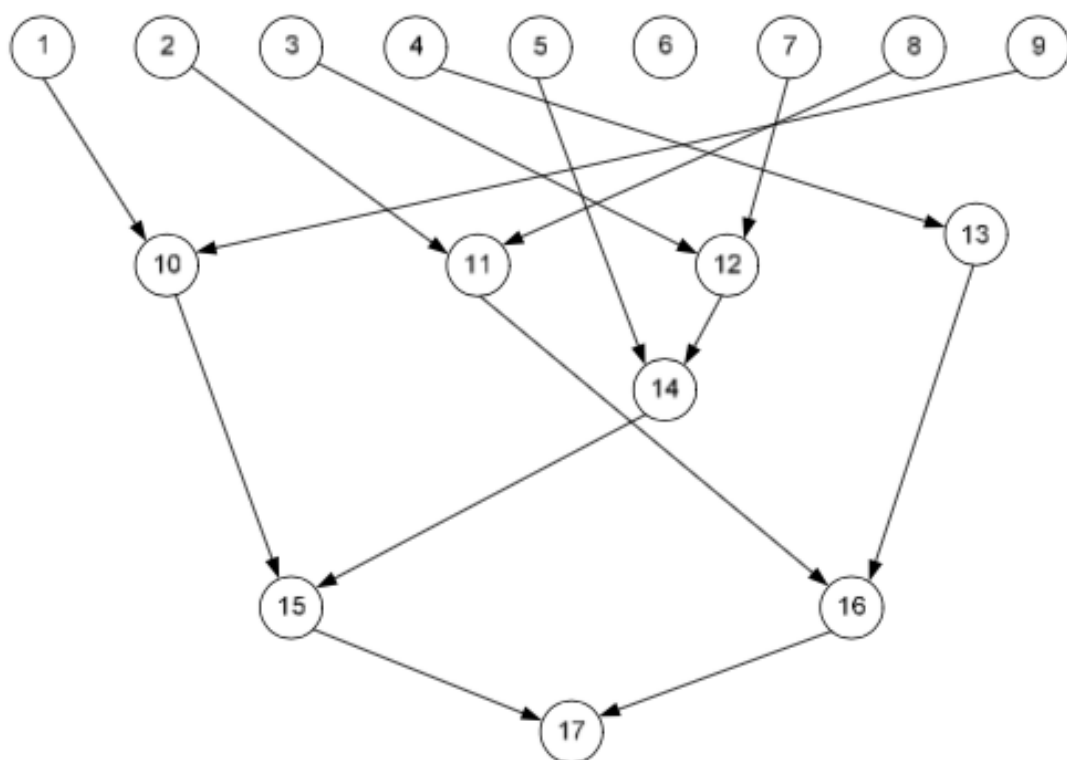


a

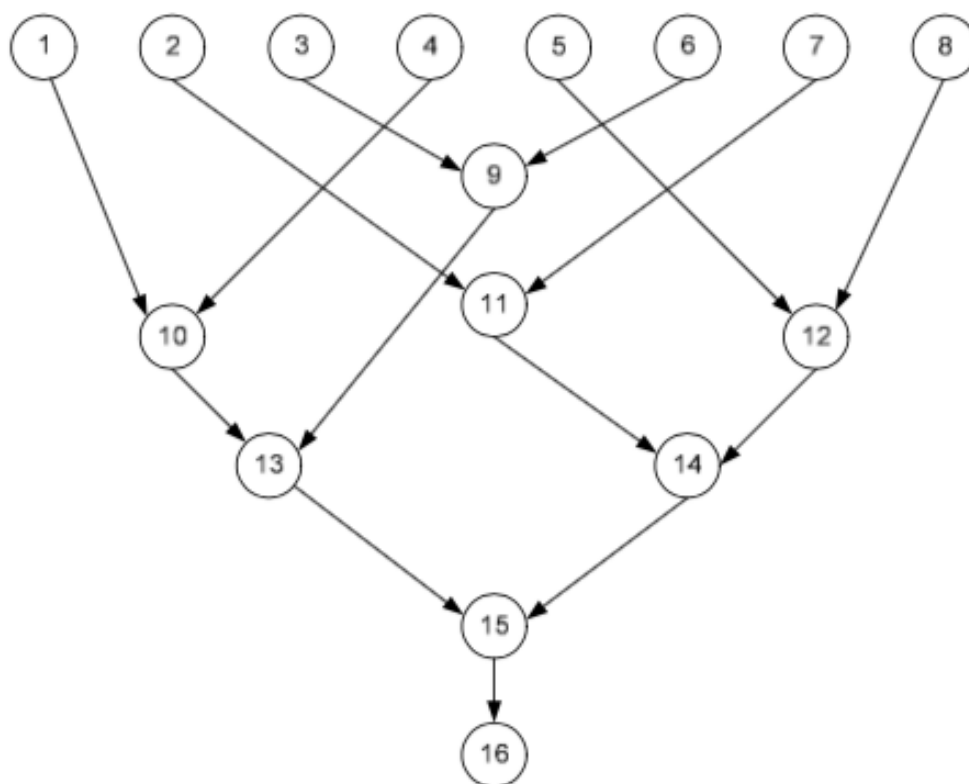


б

Рисунок 7.7 – Граф обчислювальної задачі: а – остання цифра залікової книжки 2; б – остання цифра залікової книжки 3

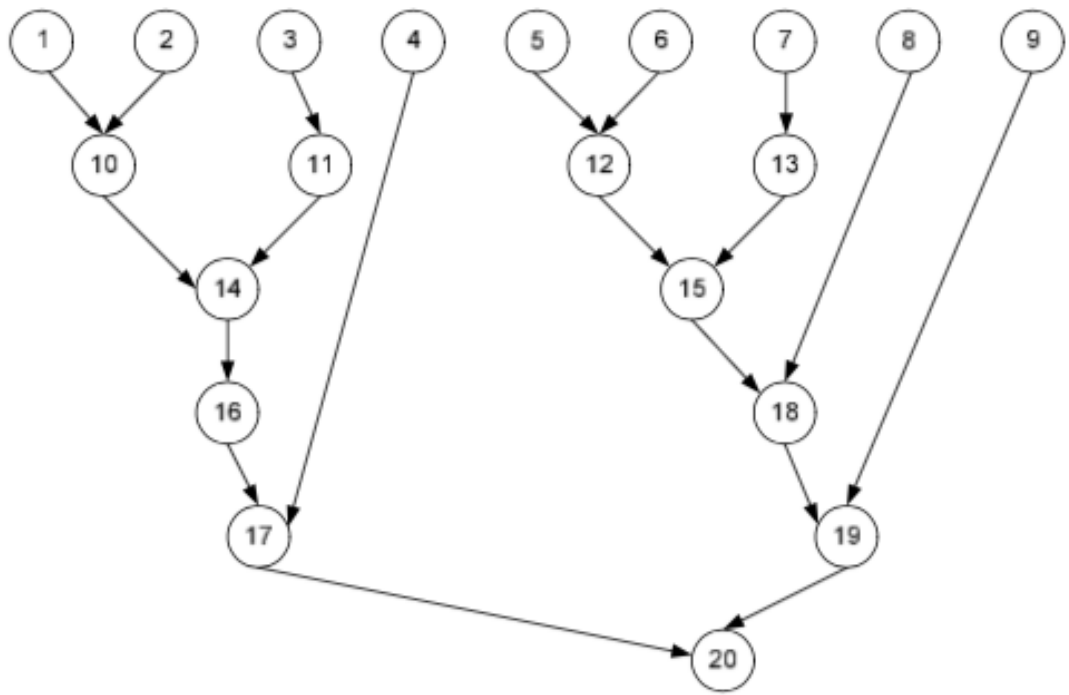


a

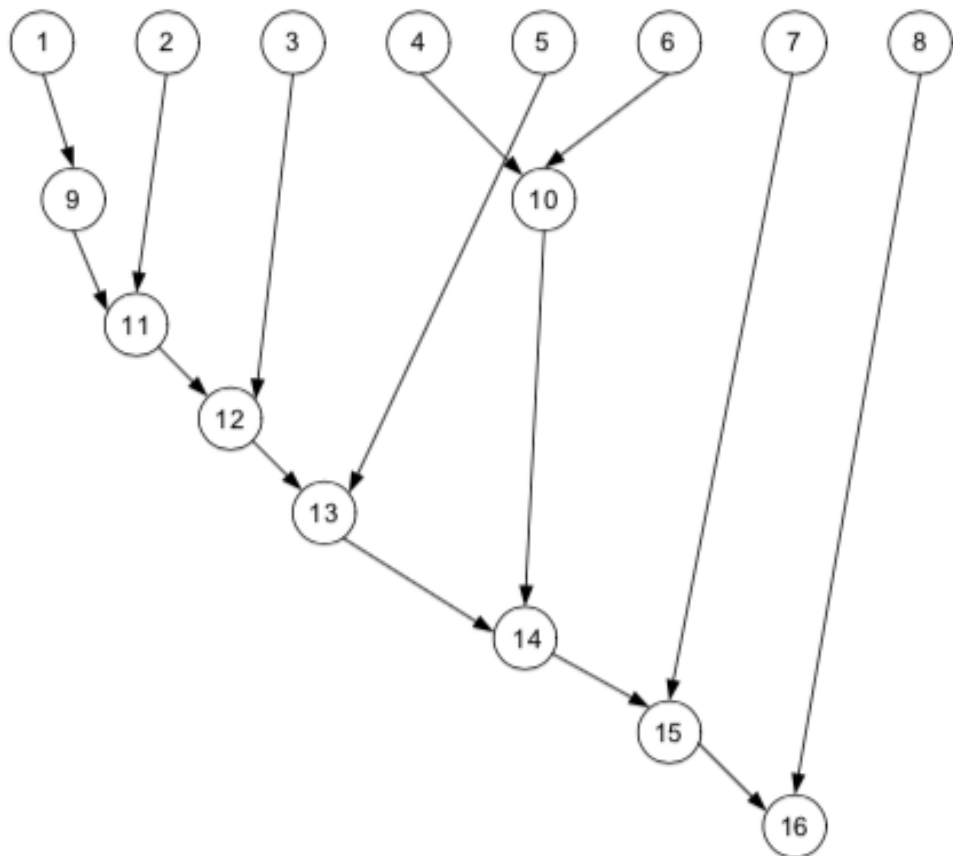


б

Рисунок 7.8 – Граф обчислювальної задачі: а – остання цифра залікової книжки 4; б – остання цифра залікової книжки 5



a



б

Рисунок 7.9 – Граф обчислювальної задачі: а – остання цифра залікової книжки 6; б – остання цифра залікової книжки 7

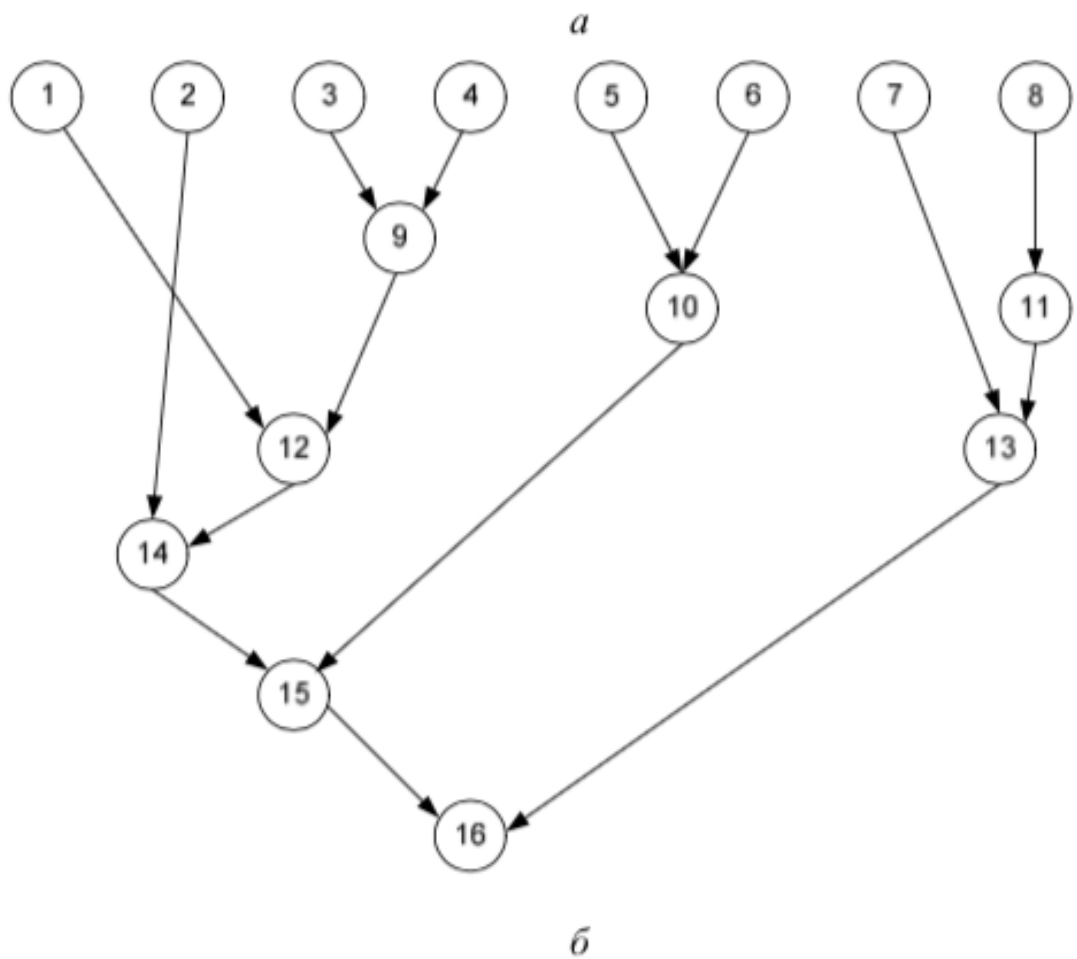
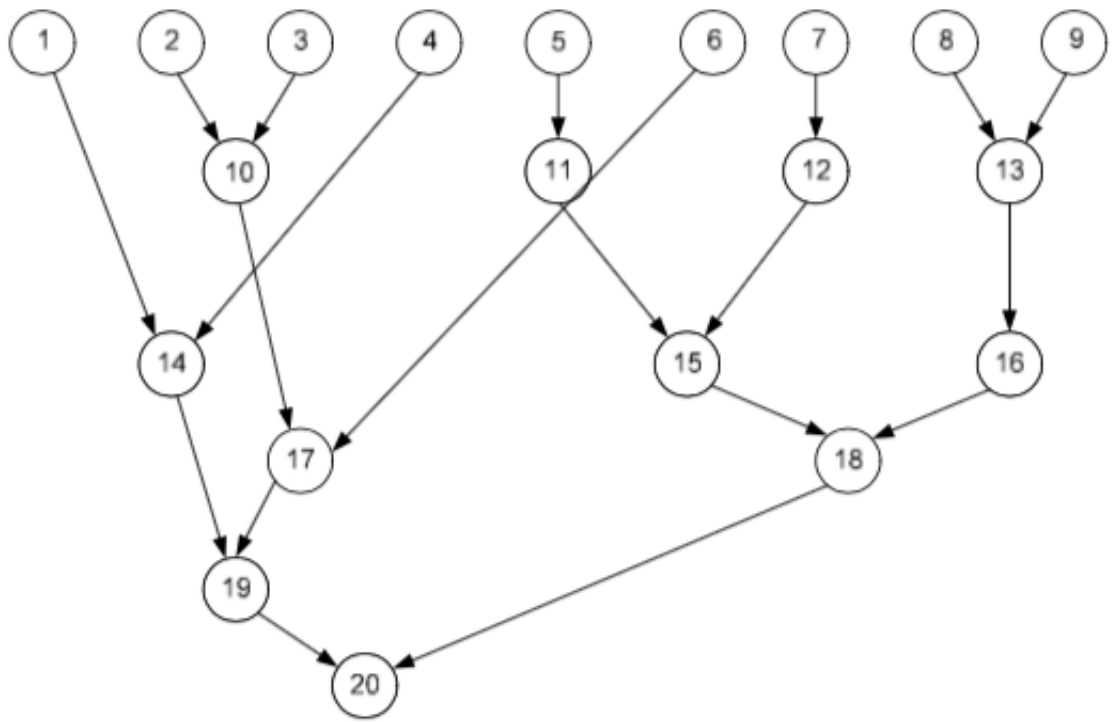


Рисунок 4.10 – Граф обчислювальної задачі: а – остання цифра залікової книжки 8; б – остання цифра залікової книжки 9

Хід роботи

1. Вивчити теоретичні відомості.
2. Отримати допуск до роботи.
3. Вибрати за методичними вказівками варіант завдання.
4. Нарисувати граф обчислювальної задачі та визначити оптимальну кількість процесорів, яка потрібна для її вирішення.