

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

ПРОЕКТУВАННЯ
КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ

Опорний конспект лекцій
(спеціальності 6.050201 – системна інженерія)

ТЕРНОПІЛЬ
ТНЕУ

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ ПРОЕКТУВАННЯ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ

1.1. Теорія та методологія побудови систем

Поняття системи визначається, перш за все, виходячи з поняття "об'єкту пізнання". В.Д. Могилевський у книзі "Методологія систем" (1999р.) трактує ці поняття у взаємозв'язку наступним чином:

"Об'єкт пізнання потрібно розглядати як систему, яка функціонує у середовищі взаємозв'язків своїх компонентів і взаємодіє з іншими системами". Цей підхід знайшов своє вираження у теорії систем та її прикладному аспекті – системному аналізі.

Теорія систем народилася як деякі узагальнення кібернетики шляхом поширення її ідей на складні організації компонентів систем. В той же час теорія систем не дає конкретних результатів для практичного застосування, в той час, як системний аналіз знайшов широке застосування в технічних задачах.

Такі атрибути, як складність, поліфункціональність, зв'язок з зовнішнім середовищем, прийняття рішень, самоорганізація і вдосконалення та управління складають зміст базисних понять теорії систем.

Класифікація систем визначається існуванням природних та штучних (технологічних та духовних) систем (рис.1.1).

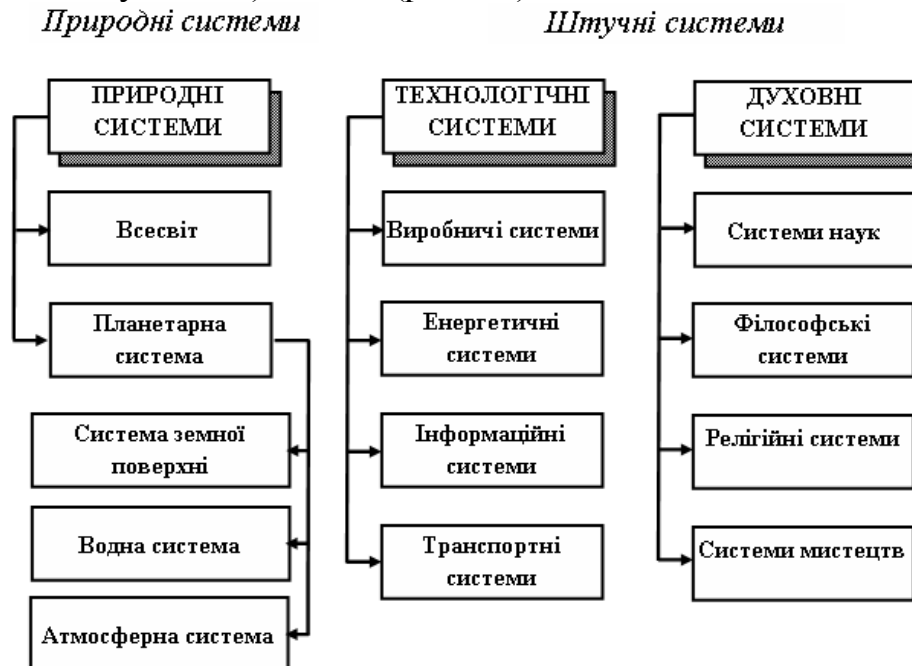


Рис.1.1. Загальна класифікація систем

В основі системи повинен бути деякий матеріальний продукт, який в сучасному розумінні включає в себе простір, матерію, енергію та інформацію. Для успішного функціонування таких надсистем виникає множина обслуговуючих їх вторинних та допоміжних класів систем. Таким чином, створюється ієрархія

систем. Структура системи при цьому визначається сформульованою ціллю її функціонування. Надлишковість системи визначається умовою її надійності, тобто підсистеми, які не відповідають функціям системи, відмирають та ліквідуються. Елементи системи повинні бути інформаційно, матеріально та енергетично пов'язані між собою для досягнення глобальної цілі системи. Системі притаманна властивість розвиватися, адаптуватися до нових умов шляхом створення нових зв'язків та компонентів.

Таким чином: система є особлива організація спеціалізованих елементів, об'єднаних в єдине ціле для рішення конкретної задачі.

Місце існування природних систем – біосфера, штучних систем – ноосфера, яка охоплена розумною людською діяльністю.

Важливим класом штучних систем є інформаційні системи, які складають основу вивчення даного предмету, який охоплює задачі проектування їх вузького класу, а саме інформаційних спеціалізованих комп'ютерних систем.

Методологія вивчення систем.

Важливим кроком у побудові теорії систем є розробка аксіоматики, яка характеризує область і умови застосування теорії, а також вибір математичного апарату досліджень в галузі проектування та аналізу систем. По-перше, слід визначити термін – абстрактна теорія систем, яка відокремлює теорію від прикладних задач і формалізує математичний апарат, який відповідає методології побудови систем.

Абстрактна теорія систем нараховує 8 рівнів:

- символічний або лінгвістичний;
- теоретико-множинний;
- абстрактно-логічний;
- топологічний;
- логіко-математичний;
- теоретико-інформаційний;
- динамічний;
- евристичний.

Принциповими особливостями системи є: динамізм – змінність її станів у часі; інформаційність взаємодії її компонентів.

Вказані рівні та особливості систем складають фундаментальні основи теорії штучного інтелекту.

Можливість реалізації певних властивостей в системі базується на функціонуванні спеціалізованих елементів її архітектури. При цьому необхідна інформаційна взаємодія між елементами, які реалізуються за допомогою каналів зв'язків та сигналів, що визначає інформаційну властивість системи. Аналіз інформаційних властивостей системи тісно пов'язаний з її складністю.

Найважливішою функцією системи є її стійкість. Порушення стійкості роботи системи пов'язане з поняттям рівноваги, яка визначається впливом її параметрів (рис.1.2).

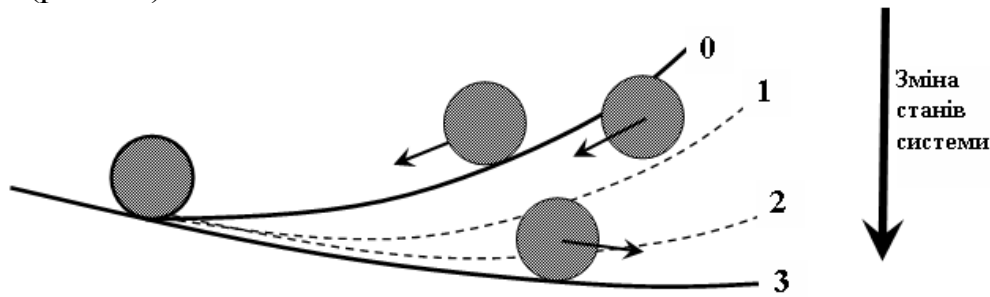


Рис.1.2. Вплив зміни параметрів на стійкість.

Втрата стійкості системи може відбуватися через зміну параметрів системи (біфоркація), через наявність непередбачених при створенні системи зовнішніх впливів, а також при порушенні інформаційних зв'язків в системі, коли змінюється її структура. В останньому випадку мова йде про структурну нестійкість.

Найбільш загальним представленням структури системи є оргграф взаємозв'язків між її компонентами, прикладом якої є схема формування математичної моделі системи (рис.1.3).

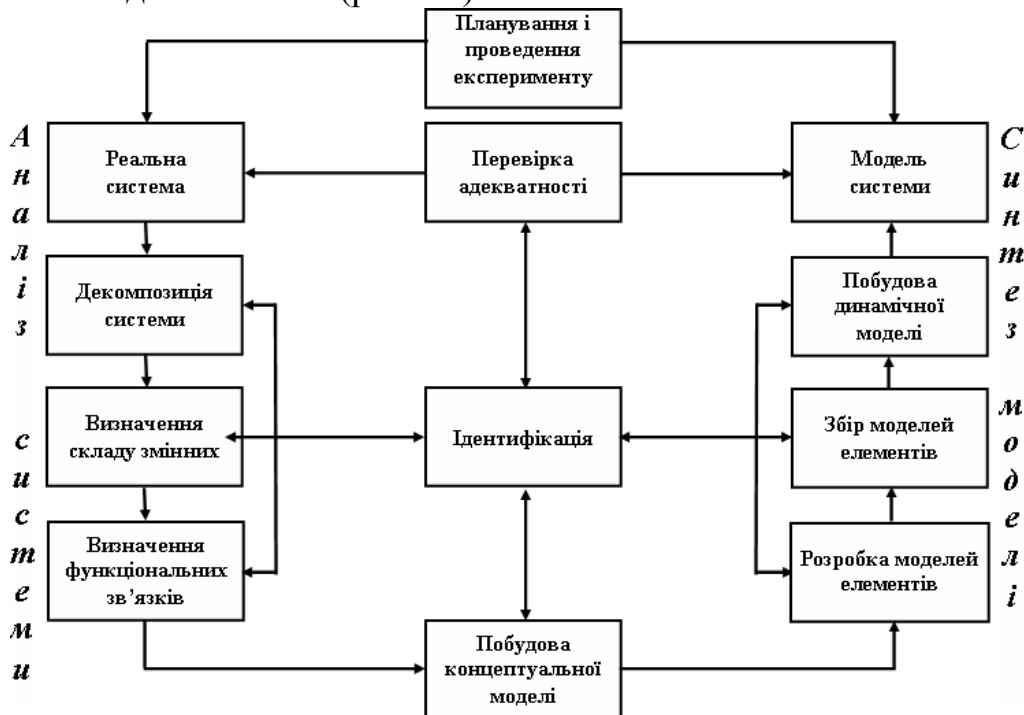


Рис.1.3. Схема формування математичної моделі системи.

Важливим атрибутом системи є її модель, яка має наступне визначення. Моделлю називається спеціально синтезований для досліджень об'єкт, який характеризується необхідним ступенем подібності до досліджуваного об'єкта та

адекватний цілям дослідження та характеристикам. При цьому при побудові моделі системи спочатку створюється мікро модель її окремих елементів, на основі яких будується макромодель або глобальна модель системи. В математиці моделлю називають деяку множину з заданим на ній набором відношень. В якості елементів множення виступають елементи системи, а відношеннями є зв'язки між ними.

Важливим класом моделей є концептуальні моделі. Вони описують перетворення інформації в системі і процеси її циркуляції по каналах зв'язку, тобто класифікуються як моделі руху даних у системі. В теорії систем під рухом розуміють зміну станів, обумовлену зовнішніми та внутрішніми причинами. Концептуальні моделі є інструментом кількісного пізнання функцій системи, як множини з заданими на ній відношеннями.

Важливою характеристикою відображення моделі системи є її структура, під якою розуміють сукупність її функціональних елементів, об'єднаних зв'язками. При цьому в якості зв'язків можуть виступати матеріальні, енергетичні та інформаційні потоки.

Елементи є складовими частинами будь-якої S_0 . Кожен з елементів також може бути S_j , тобто

$$S_0 = F(S_1, S_2, \dots, S_j, \dots, S_k),$$

де k – число підсистем.

Елементи, які вступають та виходять із системи, відповідно називаються вхідними та вихідними.

Процес перетворення.

В системах відбуваються процеси перетворення, у результаті яких змінюються стани системи, а вхідні елементи трансформуються у вихідні. В організованій системі цінність і корисність вхідних елементів збільшується, у протилежному випадку система деградує.

Вхідні елементи і ресурси.

Вхідні елементи інформаційної системи – це сигнали та вхідні інформаційні потоки. Ресурсами при цьому виступають математичні засади алгоритми та програми формування, формалізації, перетворення, передавання, зберігання, архівації та використання даних.

Вихідні елементи.

Вихідні елементи являють собою результати процесів перетворення в системі і розглядаються, як певна доцільність або прибуток від функціонування системи.

Ознаки системи та її компонентів.

Системи, підсистеми та елементи можуть мати "кількісні" та "якісні" ознаки, які тісно пов'язані з теорією вимірювань.

Навколишнє середовище системи.

Навколишнє середовище, з яким взаємодіє система, визначається певними границями по відношенню до ресурсів та споживачів системи.

На рис.1.4 показана схема взаємодії системи з навколишнім середовищем.

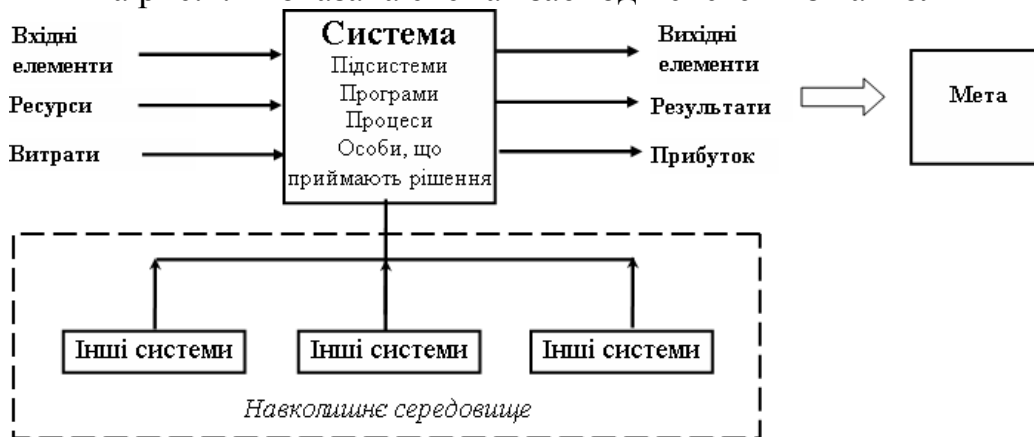


Рис.1.4. Система, її входи–виходи і навколишнє середовище.

Задачі і цілі.

При проектуванні систем першочергове значення має визначення завдань та цілей. Міра ефективності системи показує, наскільки досягаються цілі системи у процесі її функціонування.

Компоненти, завдання, програми.

У ціленаправлених системах процес перетворення організується на основі використання функцій компонентів, завдань та програм.

Прийняття рішень.

Дії і рішення, які мають місце в системі, є прерогативою керуючих підсистем або елементів, які організують систему на досягнення поставлених цілей, а також контролюють та діагностують регламентні стани системи.

Структура системи.

Поняття структури пов'язано з упорядкуванням відношень, які зв'язують елементи системи. Структура може бути простою або складною, однорівневою та ієрархічно-багаторівневою. Оцінкою функціональних ресурсів структури системи є емерджентність, яка виражається відношенням числа зв'язків N_3 до числа елементів N_0 системи.

$$k_e = \frac{N_3}{N_0} \geq 2.$$

Стани та потоки.

Існує різниця між станами і потоками в системах. Стан системи характеризується значеннями ознак системи у певний момент часу.

Переходи елементів системи з одного стану у інший викликають потоки, які визначають швидкість зміни ознак системи та її станів.

Поведінка системи.

Поведінкою системи вважається зміна станів системи у часі.

При проектуванні систем особливої уваги заслуговують чотири важливі проблеми:

- визначення границь системи та границь навколишнього середовища, з якими взаємодіє система;
- встановлення чітких цілей системи;
- визначення структури програми та алгоритмів функціонування системи;
- опис управління системами.

Границі навколишнього середовища системи поширюються і охоплюють тільки ті компоненти, які контролюються і керуються системою. Проблема встановлення границь системи нерозривно пов'язана з визначенням цілей системи та вибором критеріїв ефективності. При цьому можлива множина суперечливих цілей і різних критеріїв оцінки ефективності системи.

Структура програми – це подання відношень всіх елементів у відповідності до функцій системи. Можливо визначити структуру програми як блок-схему залежностей між різними елементами або як можливі шляхи досягнення цілей системи.

Термін управління є загальним поняттям, яке включає всі дії компонентів системи, приймаючих рішення.

Рівні системи.

Розрізняють наступні рівні систем:

- система високого рівня;
- система;
- підсистема.

Система високого рівня є керуючою по відношенню до системи, а підсистема може бути залежною, але автономною системою, яка має свою ціль.

Зворотні зв'язки.

У системах обов'язково повинні існувати додатні або від'ємні зворотні зв'язки, які забезпечують необхідну стійкість системи.

Організація системи.

Організація є функціональною характеристикою системи, яка не тотожна складності її структури.

Математичні аспекти теорії систем.

Дві системи можуть бути названі подібними на основі оцінки ступеня близькості їх математичних моделей. Системи вважаються тотожними, якщо їх математичні структури моделей ізоморфні.

Рівні вивчення систем згідно концепції Боулдинга.

1. Емпіричний, коли знання про систему отримують тільки у результаті безпосереднього спостереження.

2. Формальний, на якому реалізуються проекти систем та інструкції їх функціонування.

3. Аванпроектний, коли знань достатньо для попередньої розробки проекту системи.

4. Теоретичний, на основі якого визначаються суттєві властивості системи.

5. Системно-теоретичний, коли визначені узагальнені характеристики систем.

Складність систем.

Складність є продуктом об'єму та руху інформаційних потоків у системі.

Самоорганізація системи.

Система, яка формує під дією зовнішніх впливів у результаті свого розвитку нову систему, яка відрізняється від початкової.

1.2. Комп'ютерні системи.

Комп'ютерні системи можуть належати до двох класів: у залежності від способу опрацювання даних та інформаційних потоків:

- вбудовані КС;
- концентровані КС;
- розподілені КС пакетного опрацювання даних;
- розподілені КС реального часу.

Вбудовані КС, як правило, реалізуються на базі однієї обчислювальної платформи, мікропроцесора або мікроконтролера. При цьому всі обчислювальні процеси відбуваються в межах кристалу процесора та кристалів периферійного обладнання, яке за допомогою фізичних електронних чи оптоелектронних звуків взаємодіє з базовим процесором.

КС найчастіше використовуються в КСУ мобільних та автономних об'єктів управління. Приклад архітектури вбудованої КС показано на рис.1.5.

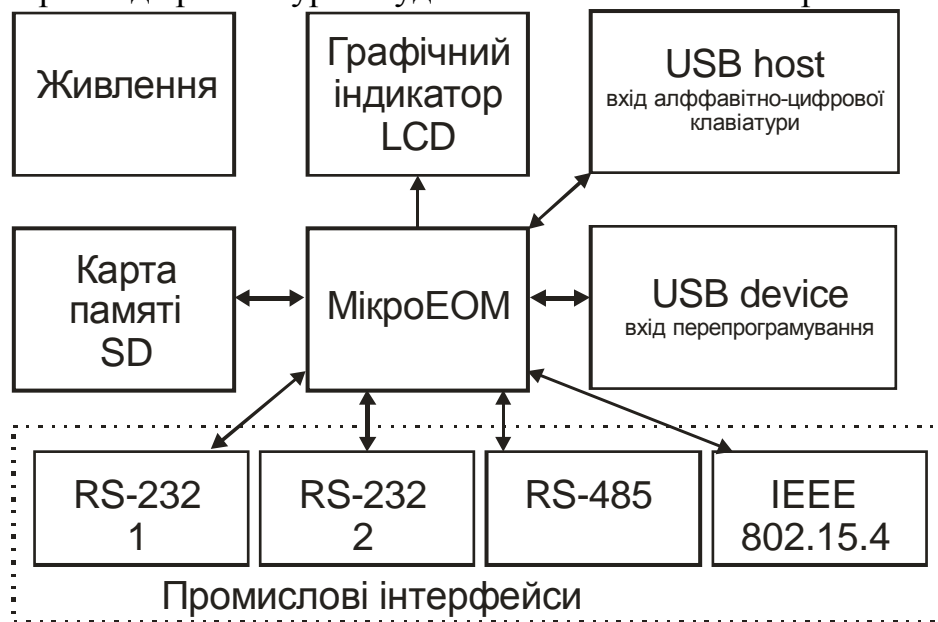


Рис.1.5. Архітектура вбудованої КС.

Концентровані КС реалізуються на основі потужного процесора і багатьох терміналів, які на невеликій віддалі, наприклад, в межах однієї технологічної

установки, машини, літака, судна або приміщення та офісу, взаємодіють через фізичні електромагнітні чи оптичні лінії зв'язку з центральним процесором. Архітектура КС концентрованого опрацювання даних показана на рис.1.6.

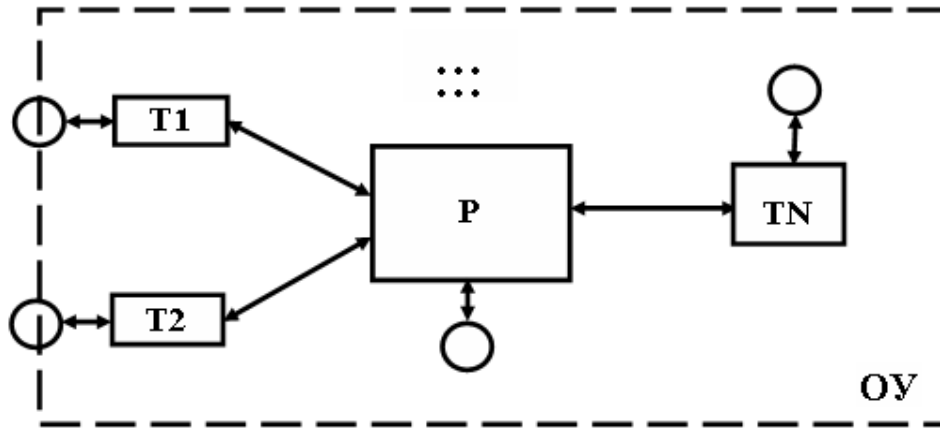


Рис.1.6. Архітектура концентрованої КС.

Особливістю таких КС є використання засобів високошвидкісного паралельного та послідовного інтерфейсів, а також автономних сенсорів.

Розподілені КС (РКС) реалізуються на основі обчислювальних мереж та віддалених процесорів-сателітів, які обслуговуються та інформаційно взаємодіють з одним або багатьма системними серверами (СС). РКС можуть мати різні архітектури, які відображаються узагальнено моделлю на рис.1.7.

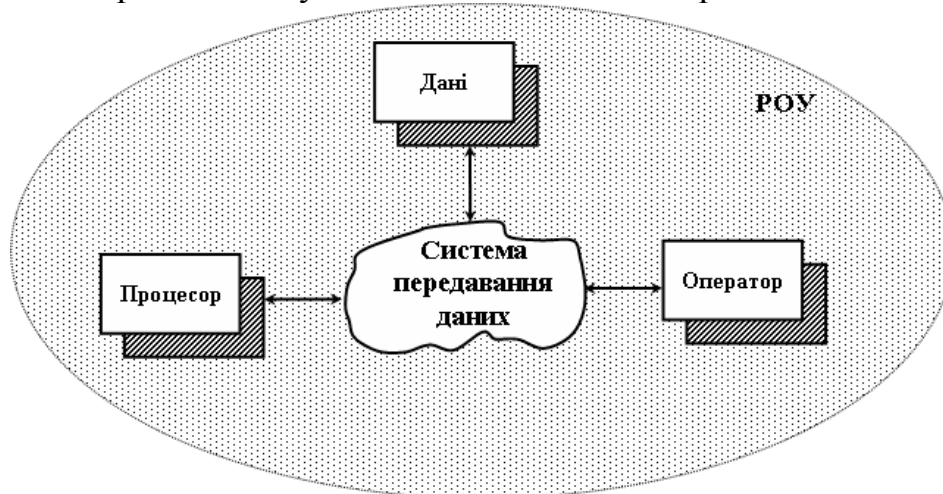



Рис.1.7. Узагальнена архітектура РКС (РОУ – розподілений об'єкт управління;  – ознака багаточисельності).

Розподілені КС реального часу можуть належати двом класам:

- універсальні РКС;
- спеціалізовані РКС або спеціалізовані КС (КСУ).

Універсальні РКС також можуть бути реалізовані на основі різних архітектур та топологій. Глобальна модель таких систем показана на рис.1.8.

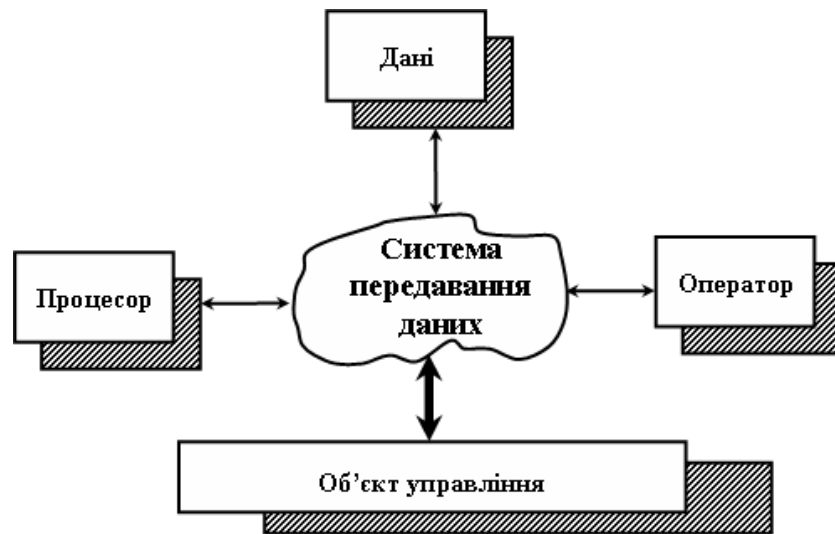


Рис.1.8. Глобальна модель РКС.

З рис.1.8 видно, що в інформаційну структуру РКС та КСУ, крім процесорів, даних, системи передавання даних та операторів, обов'язково входить віддалений розподілений об'єкт управління.

Відмінність КСУ від універсальної РКС полягає в тому, що в КСУ особливі гіперхарактеристики. Наприклад, в залежності від умов експлуатації та необхідних характеристик обладнання КСУ (високі або низькі температури, вібрації, вибухонебезпечність, мінімальні габарити, особливо висока надійність, максимальна швидкодія, об'єм пам'яті та інш.), а також унікальних особливостей об'єкта управління, КСУ максимально адаптується програмно-апаратними засобами до характеристик, необхідних алгоритмів та процесів опрацювання інформаційних потоків у реальному часі. При цьому особлива увага надається реалізації значного числа інформаційних моделей ДІ та ОУ, які оперативно використовуються для контролю відхилень станів ОУ від норми, виявлення передаварійних та попередження аварійних станів ОУ, а також обслуговуванню інформаційних потоків у квазістаціонарних станах ОУ.

1.3. Спеціалізовані комп'ютерні системи.

Спеціалізовані комп'ютерні системи (КСУ) відрізняються від універсальних та інших класів комп'ютерних систем умовами максимізації або мінімізації системних характеристик їх компонентів:

Виходячи з ресурсних характеристик системних об'єктів глобальної моделі КСУ (рис.1.8), які описуються узагальненим функціоналом

$$\hat{A} = F(T, V, M, S),$$

де T – час використання ресурсу, V – швидкість виконання системних операцій (формування, передавання, цифрова обробка та зберігання даних), M – об'єм використовуваного ресурсу пам'яті, S – системні функції.

До характеристик кортежу функціоналу, на відміну від універсальних комп'ютерних систем, висуваються особливі мінімаксні вимоги наступного виду:

$$T = \min \vee \max;$$

$$V = \min \vee \max;$$

$$M = \min \vee \max;$$

$$S = \min \vee \max .$$

Вказані вимоги визначають базові параметри компонентів КСУ в залежності від класу, характеристик стаціонарності, екологічної безпеки, вибухонебезпечності, особливих умов експлуатації (великі глибини в морі, підземні об'єкти, космічні, стратегічні та ін. об'єкти).

Найчастіше одночасне досягнення суперечливих умов до системних характеристик компонентів КСУ є в принципі неможливим або недоцільним. Тому задачі проектування та створення ефективних в архітектурному, функціональному та інформаційному аспектах є багатоваріантними з нечітко вираженими оптимумами, що потребує диференціації проектних рішень, стратегій проектування та побудови значного числа інформаційних моделей об'єктів управління, що обслуговують КСУ.

Спеціалізовані комп'ютерні системи (КСУ) належать до класу проблемно-орієнтованих комп'ютерних систем, вбудованих та розподілених комп'ютерних систем (РКС) реального часу, а також суперфункціональних комп'ютерних систем (КС) спеціального призначення та відповідного програмно-апаратного виконання.

Проектні та експлуатаційні характеристики КСУ визначаються не умовами універсальності та масового тиражування, а найчастіше особливими властивостями та характеристиками об'єктів управління, які вони обслуговують. КСУ повинні володіти відповідними унікальними характеристиками алгоритмічної, структурної, ємнісної, апаратної складності, надійності та живучості в особливих умовах, часто, критичних чи жорстких обмеженнях на системні параметри часу, швидкодії, об'єму пам'яті та операційного інтелекту.

Слід мати на увазі, що реальні ситуації створення КСУ найповніше характеризуються двома особливостями.

Перша особливість полягає в тому, що саме КСУ є своєрідним засобом апробації нових методів автоматизації обробки інформації, які мають математичні корені. Наприклад, розпаралелювання та децентралізація обчислень, макрооперації та використання функціональних розширювачів, символічна обробка та розв'язання задач у багатовимірних числових системах пройшли спочатку дуже ретельну перевірку в КСУ і тільки після цього з'явилися в УКС.

Друга особливість пов'язана з тим, що реальні КСУ є складними програмно-технічними комплексами, в яких необхідно задовільнити багато суперечливих вимог. Тому досягнення оптимальних функціональних якостей КСУ може бути проблематичним, а тому доцільніше визначити ці якості як оптимізовані, тобто

такі, що тією чи іншою мірою наближаються до оптимальних. Аналіз математичних методів оптимізації КСУ показує, що вони сприяють певною мірою виявленню недоліків таких систем і їхні «слабкі місця», простежувати взаємозв'язок характеристик системи, визначати загальний напрямок підвищення їх ефективності та оцінювати різні варіанти КСУ. Однак ці методи, як правило, не дають конструктивних розв'язків і шляхів удосконалення КСУ, не визначають змістовної сторони різних варіантів їх організації і реалізації. Генезис таких варіантів формальними математичними методами практично неможливий. Тому процес створення оптимізованих КСУ має характер багатоступеневої ітераційної процедури, де в різних відношеннях комбінуються формальні та конкретно-змістовні методи, що відіграють аналітичну (оціночну) та синтетичну (генеративну) роль.

Отже, КСУ – це КС для розв'язання багатьох відносно вузьких класів задач, оптимізовані в певній критеріальній сукупності.

На сьогодні в теорії КСУ сформувалися два основні підходи до організації їх структур. Перший ґрунтується на використанні стандартних універсальних процесорів, а врахування особливостей виконуваних задач реалізується, по суті, шляхом спеціалізації системного програмного забезпечення. Другий підхід базується на використанні процесорів, орієнтованих на виконувани алгоритми (функції) апаратним способом. Такі процесори називають функціонально – або апаратно-орієнтованими (в англійській літературі – ASIC, ASIP).

Основна перевага процесорів універсальної архітектури – гнучкість. Запис програм виконання заданого набору алгоритмів у пам'ять програм дає можливість створити спеціалізований процесор із заданими функціями. Такі процесори підлягають перепрограмуванню заміною вмісту пам'яті програм. При створенні відповідних трансляторів програмне забезпечення КСУ може бути написане мовами програмування високого рівня, що робить їх доступними для широкого кола користувачів. Крім того, дуже суттєвою перевагою цього підходу є можливість використання створених раніше програмних засобів. Разом з тим, існує ряд причин, через які використання стандартних процесорів універсальної архітектури для побудови КСУ може бути недоцільним.

По-перше, це висока трудомісткість розробки, оскільки до складу КСУ, крім самого універсального процесора, необхідно включити також засоби для реалізації інтерфейсних функцій, синхронізації, розширення пам'яті програм і пам'яті даних тощо. Сам процес розробки вимагає створення необхідного програмного забезпечення, наявності або створення технологічних програмних засобів для відпрацювання програм, а також програмно-апаратних засобів для відлагодження апаратної та програмної частин спеціалізованих процесорів.

По-друге, універсальна архітектура може бути дуже надлишковою у функціональному і структурному відношеннях для розв'язання однієї конкретної задачі (наприклад, для перетворення координат у системах орієнтації). Це може

привести до надлишкової споживаної потужності, збільшення ступеня інтеграції, кількості виводів корпусів та розмірів кристалів.

По-третє, універсальний процесор може не задовольнити вимог щодо продуктивності КСУ. Забезпечити необхідну продуктивність можна шляхом побудови багатопроцесорних КСУ. Однак їх використання може бути занадто дорогим. Тут мається на увазі вартість не стільки власне багатопроцесорної системи, скільки допоміжних засобів, що забезпечують використання такої системи. Особливо багато проблем пов'язано з розробкою паралельних обчислювальних процесів, при реалізації яких всі процесори були б повністю завантажені. В такому разі системне програмне забезпечення має бути здатним оперативно розв'язувати задачу оптимізаційного планування завантаження процесорів багатопроцесорної системи конкретної архітектури. Такий підхід може вимагати великих витрат обладнання, що є небажаним (особливо при використанні КСУ у бортових системах).

Зараз, коли досягнення мікроелектронної технології підтримується потужними САПР, другий підхід, що передбачає створення апаратно-орієнтованих на виконувани алгоритми процесорів, є реальною альтернативою універсальним стандартним процесорам.

По-перше, такий підхід забезпечує максимально можливу продуктивність при розв'язанні заданої задачі.

По-друге, він вимагає мінімальних витрат обладнання на побудову КСУ для розв'язання заданої задачі за рахунок компромісу між програмними та апаратними засобами. Таке спільне проектування апаратної і програмної частин КСУ дістало назву hardware/software co-design.

По-третє, сучасні мови опису апаратних засобів, зокрема VHDL, мають настільки високий рівень, що процес проектування апаратно-орієнтованого процесора не є набагато складнішим у порівнянні з розробкою спеціального програмного забезпечення для універсального процесора. Якщо врахувати ще й можливості сучасних САПР щодо забезпечення повного відпрацювання моделей електронних компонент, включаючи їх роботу в складі КСУ, а також використання бібліотек раніше створених електронних компонент, то стає очевидним, що настав час, коли другий підхід починає витісняти перший.

З викладеного випливає, що розвиток КСУ в Україні зумовлений, з одного боку, потребами в КСУ, а з іншого – можливостями їх власної розробки і виробництва. Світовий досвід створення КСУ підтверджує життєздатність цього напрямку комп'ютерної науки і техніки в сучасних умовах, незважаючи на імпортозне походження значної частини мікроелектронних комплектуючих виробів. Забезпечення необхідного кваліфікаційного рівня кадрового супроводу такого розвитку має бути однією з найважливіших задач національної вищої технічної школи.

Контрольні запитання і завдання.

1. Розкрийте структуру загальної класифікації систем.
2. Назвіть рівні абстрактної теорії систем.
3. Охарактеризуйте найважливішу функцію систем – стійкість.
4. Представте структуру формування математичної моделі системи.
5. Опишіть вхідні та вихідні характеристики взаємозв'язку системи з навколишнім середовищем.
6. Розкрийте поняття емерджентності системи.
7. Назвіть рівні вивчення систем згідно концепції Боулдинга.
8. Охарактеризуйте класи комп'ютерних систем.
9. Представте архітектуру концентрованої КС.
10. Охарактеризуйте архітектуру розподіленої КС.
11. Охарактеризуйте системні об'єкти глобальної моделі КСУ.
12. Поясніть відмінності та критерії побудови КСУ.

2. СИСТЕМНІ ОБ'ЄКТИ, СИСТЕМНІ ФУНКЦІЇ, ФУНКЦІОНАЛЬНІ ОБ'ЄКТИ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ

Глобальна модель комп'ютерної системи, яка формально описує структуру КСУ, вперше була запропонована Николайчуком Я.М. у вигляді взаємодії п'яти типів системних об'єктів (рис.2.1).



Рис.2.1. Глобальна модель комп'ютеризованої системи.

Кожен з наведених системних об'єктів може виконувати в середовищі КС чотири системні функції:

- формування даних ;
- передавання даних;
- цифрова обробка даних;
- приймання та зберігання даних.

Тобто, кожен з системних об'єктів (СО) може бути одним з функціональних об'єктів наступного типу:

- джерело інформації (ДІ);
- середовище передавання інформації (СПІ);
- середовище цифрової обробки інформації (СОІ);
- приймач інформації (ПІ).

Отже, системні об'єкти КС характеризуються дуальними (поліфункціональними) властивостями, що в значній мірі ускладнює методологію проектування та теоретичні основи оптимізації параметрів КС.

Виходячи з класифікації п'яти системних об'єктів КС, можна побудувати таблицю пар їх взаємодії через інтерфейсні засоби комунікацій (табл.2.1).

Таблиця 2.1.

	Р	Д	СПД	ОУ	О
Р	Р→Р	Р→Д	Р→СПД	Р→ОУ	Р→О
Д	Д→Р	Д→Д	Д→СПД	Д→ОУ	Д→О
СПД	СПД→Р	СПД→Д	СПД→СПД	СПД→ОУ	СПД→О
ОУ	ОУ→Р	ОУ→Д	ОУ→СПД	ОУ→ОУ	ОУ→О
О	О→Р	О→Д	О→СПД	О→ОУ	О→О

Очевидно, що для вивчення названих інтерфейсних взаємодій СО та використання їх при проектуванні КС необхідно описати взаємодію 25-ти їхніх пар. В той же час, враховуючи, що теорія, методологія та техніка реалізації КС на основі стандартних технічних засобів, міжнародних протоколів та інтерфейсів достатньо повно подана у відповідних виданнях та інструкціях, при проектуванні низових рівнів проблемно-орієнтованих та спеціалізованих КС особливу увагу слід надавати вивченню системних характеристик об'єктів управління та їх взаємодії з іншими об'єктами КС. Дані взаємодії ОУ та СПД з іншими СО відображені в табл. 2.1 відповідним фоном. Широка різноманітність реальних ОУ (наприклад, космічні апарати, літаки, підводні та наземні човни, атомні станції, енергетичні та нафтопромислові системи, транспортні засоби, інформаційні системи, телекомунікації та інше) вимагає відповідної проблемної орієнтації та адаптації КС до характеристик ОУ при проектуванні та аналізі діючих КС. Ця адаптація потребує по - новому осмислити формалізацію опису характеристик системних об'єктів.

2.1. Формалізація опису характеристик системних об'єктів КСУ.

У загальному випадку ресурсні характеристики СО проектованої КС можуть бути достатньо повно описані функціоналом, який заданий коефіцієнтом E_{co} та четвіркою параметрів

$$\hat{A} = F(T, V, M, S), \quad (2.1)$$

де T – час використання ресурсу, V – швидкість виконання системних операцій (формування, передавання, цифрова обробка та зберігання даних), M – об'єм використовуваного ресурсу пам'яті, S – системні функції.

При цьому границі зміни параметрів задаються системою нерівностей

$$\begin{aligned} 0 \leq T \leq T_0; \\ 0 \leq V \leq V_0; \\ 0 \leq M \leq M_0; \\ 0 \leq S \leq S_0, \end{aligned} \quad (2.2)$$

де T_0 – час формування, передавання, цифрової обробки та зберігання даних, використання технічного засобу та інше, V_0 – пропускна здатність каналу зв'язку, максимальна швидкість читання/запису, максимальна частота обміну даними та інше, M_0 – максимальний об'єм пам'яті, що використовується (ОЗП, ПЗП, магнітних, оптичних та твердих копій носіїв), S_0 – максимальний ресурс системних функцій (операційні системи, пакети прикладних програм тощо).

Якщо задати нормовані границі зміни кожного параметра (2.1) в діапазоні від 0 до 1, то реалізація коефіцієнта E_{co} отримає вигляд:

$$\hat{A} = F(0.3, 0.4, 0.7, 0.2).$$

Даний коефіцієнт доцільно привести до безрозмірної форми на основі функції адитивності $E_{\hat{n}\hat{i}} = \frac{T+V+M+S}{4}$, що забезпечує діапазон зміни E_{co} в межах $0 \leq E_{co} \leq 1$ та відповідає гіпотезі про статистичну незалежність ресурсних параметрів СО (2.1). При цьому економічна собівартість руху даних може бути обчислена на основі рівняння:

$$D_{\hat{n}\hat{i}} = \hat{A}_{\hat{n}\hat{i}} \cdot D_0 - V_0, \quad (2.3)$$

або

$$D_{\hat{n}\hat{i}} = \frac{T+V+M+S}{4} \cdot D_0 - V_0, \quad (2.4)$$

де P_0, V_0 – відповідно прибутки та затрати на реалізацію функцій СО.

На рисунку 2.2 показані характеристики собівартості руху даних на рівні формалізованого опису системного об'єкту.

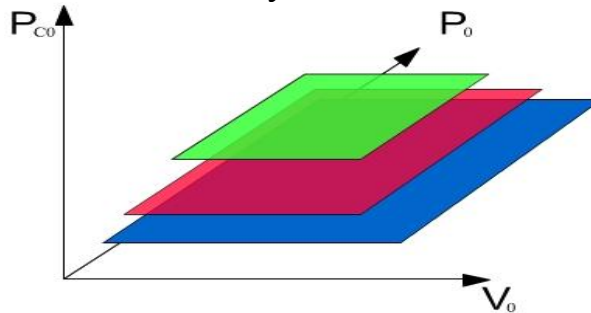


Рис.2.2. Характеристики собівартості руху даних системного об'єкту.

Аналогічні формули можуть бути застосовані для інших системних об'єктів, в яких формальні параметри S будуть відрізнятися наступними системними функціями:

P – процесор (апаратне, системне та прикладне програмне забезпечення);

\hat{A} – дані (зберігання даних в архівах, способи кодування даних, захист від помилок, захист від несанкціонованого доступу, семантичні властивості даних і т.д.);

$\hat{N}\hat{I}\hat{A}$ – мережеве програмне забезпечення, оптимізація маршрутів передавання даних, використання спецканалів, інформаційна технологія моделювання руху даних;

$\hat{I}\hat{O}$ – характеристики стаціонарності, нестаціонарності, квазістаціонарності, інформаційні технології кодування станів та контролю їхнього відхилення від норми, статистичні, кореляційні та ентропійні моделі, функції керування та побудови моделей руху даних;

\hat{I} – система знань та професійних навиків і т.д.

Функціонал характеристик системних об'єктів в багатьох випадках доцільно розширити до п'яти параметрів шляхом диференціації параметру швидкості виконання системних операцій, тобто V_R – швидкість запису (вхідний

інформаційний потік), V_W – швидкість зчитування (вихідний інформаційний потік), звідки характеристики системного об'єкта будуть описуватися параметрами:

$$\hat{A}_{\hat{n}\hat{i}} = F(T, V_R, V_W, M, S). \quad (2.5)$$

Характеристика (2.5) дозволяє врахувати асиметричність характеристик швидкодії вхідних та вихідних пристроїв системних об'єктів.

Розроблені аналітичні моделі характеристик системних об'єктів комп'ютерної мережі є базою для теоретичної формалізації руху даних. На основі вказаних моделей проводиться аналіз ефективності та розробка проектів комп'ютерних мереж з проблемною орієнтацією для конкретних технологічних процесів та підприємств.

2.2. Системні характеристики процесора.

Процесори (P) містять у своєму складі інтелектуальні, оснащені однокристальними спец процесорами, сенсори, сигнальні процесори, мікро– та міні–контролери, контролери низових мереж, комутаційні процесори, сервери та комп'ютерні кластери. Системні характеристики P достатньо повно описуються функціоналом

$$\hat{A}_P = F(T, V, M, S),$$

де T – час використання ресурсу, V – швидкість виконання системних операцій V_R/V_W , M – об'єм використовуваного ресурсу пам'яті, S – системні функції операційної системи та прикладного програмного забезпечення.

Параметр часу використання ресурсу T оцінюється у відсоткових значеннях його безперервної роботи протягом доби. Використання ресурсу даної характеристики залежить від типу та призначення процесорів. Наприклад, мікропроцесори, технологічні мікроконтролери та промислові процесори призначені для цілодобової роботи, тобто $T=1.0$. Для процесорів ЕОМ режим роботи може складати 6, 8 або 16 годин за добу, що відповідає використанню ресурсу відповідно $T_i = 0.25, 0.3, 0.6$.

Характеристика використання об'єму ресурсів пам'яті процесора M також залежить від конкретної інформаційної технології її застосування. Наприклад, мікроконтролери масового застосування на основі 8 та 16 розрядних мікропроцесорів, які оперують з обмеженою пам'яттю, використовують її ресурси практично на 100 відсотків, тобто $M_i = 1.0$. В процесорах універсального призначення, якими оснащено ПЕОМ, використання ресурсів пам'яті залежить від характеру задач, які вирішує оператор та динамічності їх реалізації в часі, тобто дану оцінку треба розраховувати на основі математичного сподівання :

$$\bar{M}_i = \frac{1}{k} \sum_{i=1}^k M_i, \quad (2.6)$$

де M_i – поточні або миттєві коефіцієнти використання ресурсів пам'яті процесора. Аналогічно може бути обчислена характеристика використання ресурсів функціональних характеристик процесора (S_i).

Приклад вхідних та вихідних інформаційних сигналів процесорів поданий в табл.2.2.

Таблиця 2.2.

Формувачі сигналів процесора

Формувачі вхідних та вихідних інформаційних сигналів процесора	Характеристики інформаційних потоків
Паралельний порт	1 Гбіт/с
Послідовний порт	100 Мбіт/с
Маніпулятор "мишка"	8 біт/с
Клавіатура	8–12 біт/с
Модем	10–100 Мбіт/с
Мікрофон	2000 * 8=біт/с 2,4–3,6 кГц
Технологічний сканер	100 Мбіт/с

Аналіз табл.2.2 показує, що швидкість створення повідомлень процесорів може змінюватися в широких межах. Діапазон швидкості обміну даними процесорів може змінюватися в межах 1–10⁹ біт/с.

2.3. Системні характеристики даних.

Сучасна систематизація даних охоплює три їх основні класи:

- фізичні дані;
- логічні дані;
- віртуальні дані.

Ресурсні характеристики СО „Дані” представлені часом зберігання, швидкістю запису, швидкістю зчитування, об'ємом та захистом даних від помилок

$$E_d = F(T, M, V_R, V_W, S),$$

де – T – час зберігання, M – об'єм, V_R – швидкість запису, V_W – швидкість зчитування, S – захист від помилок та несанкціонованого доступу.

Фізичні дані – це дані, які представлені в адресному просторі фізичних носіїв (жорстких магнітних дисків, гнучких магнітних дисків, оптичних та лазерних дисків), а також на твердих носіях (документи, таблиці, графіки).

Поняття логічних даних використовується в теорії та практиці опису логічних моделей баз даних, а також в програмних продуктах, як формальні параметри.

Віртуальні дані – це такий тип даних, які відсутні у фізичному вигляді в КС і можуть формуватися в процесі рішення задач, розрахунків, а також бути в динамічному стані при передаванні в КМ.

Способи кодування даних визначаються теоретико–числовими базисами (ТЧБ), які застосовуються для їх представлення. Найбільш широко вживаними ТЧБ в сучасних КС є наступні базиси: унітарний, Хаара, Крейга, Грея, Радемахера, Крестенсона, Уолша та Галуа, кодові матриці яких подані на рис.2.3.

$$\begin{matrix}
 M_{Uni} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{vmatrix} &
 M_{har} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{vmatrix} &
 M_{Gr} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 0 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \end{vmatrix} &
 M_{Rad} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{vmatrix} \\
 \\
 \begin{matrix}
 \text{а)} & \text{б)} & \text{в)} & \text{г)} \\
 M_{LibCr} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 \\ 0 & 0 & 1 & \dots & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{vmatrix} &
 M_{Cres} = \begin{vmatrix} P_1 & P_2 & \dots & P_n \\ 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ 2 & 2 & \dots & 2 \\ 0 & 3 & \dots & 3 \\ 1 & 4 & \dots & 4 \\ 2 & 0 & \dots & 5 \\ 0 & 1 & \dots & 6 \\ \dots & \dots & \dots & \dots \\ a_1 & a_2 & \dots & a_n \end{vmatrix} &
 M_{Gal} = \begin{vmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{vmatrix} \\
 \text{д)} & \text{е)} & \text{є)}
 \end{matrix}
 \end{matrix}$$

Рис. 2.3. Кодові матриці дискретних базисів: а) – унітарного; б) –Хаара; в) – Грея; г) – Радемахера; д) –Крейга; е) –Крестенсона; є)–Галуа.

Кожен з названих базисів характеризується визначеним об’ємом кодової матриці для представлення даних. При цьому найбільш надлишковим базисом є унітарний, в якого кодова матриця $V = N^2$, а число активних кодових елементів $n = N^2 / 2$, де N – діапазон кодування даних. Аналогічну надлишковість забезпечує базис Хаара, в два рази меншу надлишковість забезпечує базис Крейга, тобто $V = N^2 / 4$, а $n = N^2 / 8$. Максимально широке застосування для кодування даних в сучасних КС отримали базиси Радемахера та Крестенсона, в яких $V = N \log_2 N$. Дані базиси відповідно породжують двійкову систему числення та систему числення залишкових класів.

Базис Уолша максимально широко використовується в сучасних телекомунікаційних КС. Даний базис породжує систему ортогональних шумоподібних сигналів, які використовуються в сотових системах мобільного зв'язку.

Найменшу надлишковість кодування даних забезпечує базис Галуа, кодова матриця якого $V = N$, а $n = N/2$.

Згідно викладеного, характеристики ТЧБ кодування даних, як системного об'єкта, подані в табл.2.3.

Таблиця 2.3.

Характеристики потоків даних

Формувачі вхідних та вихідних інформаційних сигналів даних	Характеристики інформаційних потоків даних
Унітарний базис	$V = N^2; n = N^2/2$
Базис Хаара	$V = N^2, n = N$
Базис Крейга	$V = N^2/4, n = N^2/8$
Базис Радемахера	$V = N \cdot \log_2 N, n = \frac{N}{2} \log_2 N$
Базис Крестенсона	$V = N \cdot \log_2 N$
Базис Уолша	$V = N^2, n = N^2/2$
Базис Галуа	$V = N, n = N/2$

2.4. Характеристики моделей об'єктів управління.

Об'єкт управління адекватно може бути описаний характеристичними параметрами – часом, ентропією, моделлю об'єкта та системними функціями:

$$E_{OU} = F(T, M, I, S),$$

де T – час, M – модель об'єкта, I – ентропія, S – системні функції.

Найважливішими системними характеристиками ОУ є модель об'єкта, ентропія та системні функції.

Найважливіші моделі ОУ:

Таблиця 2.4.

Типи моделей

№	Типи моделей ОУ	Аналітичний вираз
1	2	3
1	Сигнальні аналогові	$M = X(t)$
2	Сигнальні дискретизовані квантовані	$M = X_i, i \in \overline{1, n}, 0 \leq x_i \leq A,$ де X_i – дискретизованеквантоване значення ОУ, n – об'єм вибірки, A – діапазон квантування
3	Дискретні диференціальні	$M = \Delta X_i = X_{i+1} - X_i,$ де ΔX_i – перші прирости станів ОУ.

4	Дискретні інтегральні	$M = \sum_{i=1}^k X_i,$ де k –число сумувань дискретних станів ОУ.
5	Статистичні:	
5.1	вибіркове математичне сподівання	$M_x = \frac{1}{n} \sum_{i=1}^n x_i$
5.2	ковзне математичне сподівання	$M_j = \frac{1}{m} \sum_{i=1+j}^{m+j} X_{i+j}, \quad j = 0,1,2,\dots$ де $j = 0,1,2,\dots$ –дискретний зсув;
5.3	вагове математичне сподівання	$M_v = \sum_{i=1+j}^{m+j} V_{i-j} \cdot X_{i+j},$ де V_i –вагова функція;
5.4	дисперсія	$D_x = \frac{1}{n} \sum_{i=1}^n (X_i - M_x)^2$
5.5	середньоквадратичне відхилення	$\sigma_x = \sqrt{D_x}$
6	Автокореляційні моделі	
6.1	знакова	$B_{xx}(j) = \frac{1}{n} \sum_{i=1}^n \overset{o}{\text{sign}} x_i \cdot \overset{o}{\text{sign}} x_{i+j}$ $\overset{o}{\text{sign}} x_i = \begin{cases} +1, & x_i \geq 0 \\ -1, & x_i < 0 \end{cases};$
6.2	релейна	$H_{xx}(j) = \frac{1}{n} \sum_{i=1}^n \overset{o}{x}_i \cdot \overset{o}{\text{sign}} x_{i+j}$
6.3	коваріаційна	$K_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \cdot x_{i+j}$
6.4	кореляційна	$R_{xx}(j) = \frac{1}{n} \sum_{i=1}^n \overset{o}{x}_i \cdot \overset{o}{x}_{i+j}$
6.5	нормована кореляційна	$\rho_{xx}(j) = \frac{R_{xx}(j)}{D_{xx}}$
6.6	структурна	$C_{xx}(j) = \frac{1}{n} \cdot \sum_{i=1}^n (x_i - x_{i+j})^2$
6.7	модульна	$G_{xx}(j) = \frac{1}{n} \cdot \sum_{i=1}^n x_i - x_{i+j} $

6.8	нормована модульна	$g_{xx}(j) = \frac{C_{xx}(j)}{M_x} - M_x$
6.9	еквівалентна	$F_{xx}(j) = \frac{1}{n} \cdot \sum_{i=1}^n Z_{ij}^{\vee};$ $Z_{ij}^{\vee} = \begin{cases} x_i, & x_i < x_{i+j} \\ x_j, & x_i \geq x_{i+j} \end{cases}.$
7	Взаємкореляційні моделі між двома параметрами ОУ	
7.1	взаємознакова	$B_{xy}(j) = \frac{1}{n} \sum_{i=1}^n \text{sign } x_i^o \cdot \text{sign } y_{i+j}^o$
7.2	взаєморелейна	$H_{xy}(j) = \frac{1}{n} \sum_{i=1}^n x_i^o \cdot \text{sign } y_{i+j}^o$
7.3	взаємоковаріаційна	$K_{xy}(j) = \frac{1}{n} \sum_{i=1}^n x_i \cdot y_{i+j}$
7.4	взаємкореляційна	$R_{xy}(j) = \frac{1}{n} \sum_{i=1}^n x_i^o \cdot y_{i+j}^o$
7.5	нормована взаємкореляційна	$P_{xy}(j) = \frac{R_{xy}(j)}{\sqrt{D_x + D_y}}, \text{ якщо } j = 0,$ то $P_{xy}(0)$ – нормований коефіцієнт взаємкореляції $P_{xy}(0) = \frac{R_{xy}(0)}{\sqrt{D_x + D_y}}.$
8	Взаємкореляційна матрична модель ОУ	$\begin{vmatrix} \rho_{11} & \rho_{12} & \dots & \rho_{1j} & \dots & \rho_{1m} \\ \rho_{21} & \rho_{22} & \dots & \rho_{2j} & \dots & \rho_{2m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \rho_{i1} & \rho_{i2} & \dots & \rho_{ij} & \dots & \rho_{im} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \rho_{n1} & \rho_{n2} & \dots & \rho_{nj} & \dots & \rho_{nm} \end{vmatrix},$ де $\rho_{ij} = \frac{R_{ij}(0)}{\sqrt{D_i + D_j}}$ – нормований коефіцієнт взаємкореляції між параметрами ОУ, R_{ij} – взаємкореляційна модель між i та j параметром.

Контрольні запитання і завдання.

1. Охарактеризуйте системні об'єкти та функції драйверів КСУ.
2. Охарактеризуйте системні функції об'єктів КСУ.
3. Охарактеризуйте функціональні об'єкти КСУ.
4. Поясніть взаємодію пар системних об'єктів КСУ.
5. Охарактеризуйте системні характеристики даних.
6. Назвіть основні теоретико-числові базиси..

3. ОДНОРІВНЕВІ АРХІТЕКТУРИ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ

3.1. Класифікація архітектур та характеристики КСУ.

В даний час існує широка різноманітність архітектур КСУ, до яких належать концентровані та розподілені системи опрацювання даних. До систем першого класу можна віднести монопольні архітектури, архітектури з розподіленим часом, архітектури з мультипрограмним та мультипроцесорним опрацюванням даних. Другий клас представлений значним числом однорівневих архітектур, в тому числі: магістральні, зіркові, кільцеві, систолічні. До класу багаторівневих розподілених архітектур КСУ слід віднести ієрархічні, багаторівневі–магістральні та зірково–магістральні архітектури.

Окремим класом безпроводних архітектур представлені безпроводні радіотехнічні інформаційні системи та комп'ютерні мережі наступного типу:

- безретрансляторні;
- з пасивними ретрансляторами;
- з активними ретрансляторами, в тому числі сотові мережі.

Комп'ютерні системи з оптичними каналами зв'язку охоплюються архітектурами на основі:

- дуплексних оптичних ретрансляторів;
- оптичних активних ретрансляторів;
- оптичних сканерів;
- волоконно–оптичних ліній зв'язку.

Значною оригінальністю архітектур характеризуються спеціалізовані комп'ютерні системи (КСУ), які часто можуть базуватися на об'єднанні окремих елементів різних типових архітектур. До такого класу інформаційних систем, наприклад, належать:

- системи обліку витрат енергоносіїв з глибоким розпаралеленням потоків даних;
- комп'ютерні розподілені системи екологічного моніторингу;
- спеціалізовані охоронні системи;
- проблемно–орієнтовані корпоративні системи промислових та адміністративних організацій.

Така велика кількість архітектур інформаційних систем в значній мірі ускладнює вирішення задач оптимізації проектних рішень при побудові інформаційних систем, що потребує розробки відповідних моделей архітектур, які б дозволили шляхом формалізації структурних елементів різних мереж з єдиних позицій провести дослідження та порівняння їх системних характеристик. Одним з перспективних підходів до вирішення такої задачі є використання теорії та технології побудови одномірних та багатомірних матричних моделей руху даних, що визначає актуальність таких досліджень.

На рис.3.1 подана класифікація архітектур КСУ з фізичними лініями зв'язку.



Рис.3.1. Класифікація архітектур КСУ з фізичними лініями зв'язку

На рис.3.2 подана класифікація архітектур КСУ з безпроводними лініями зв'язку.

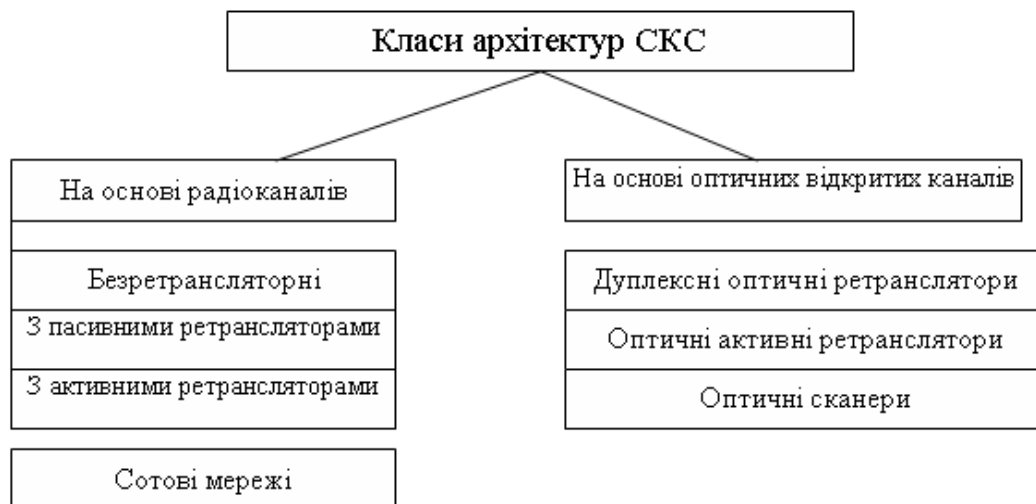


Рис.3.2. Класифікація архітектур КСУ з безпроводними радіоканалами.

Подані класифікації архітектур КСУ дозволяють виконати формалізований вибір відповідного класу архітектури КСУ в залежності від їх проблемної орієнтації та необхідних системних характеристик.

3.1.1. Системи концентрованого опрацювання інформаційних потоків.

3.1.1.1. Монопольна КСУ

Монопольна архітектура (рис.3.3) характеризується максимальним паралелізмом руху даних, внаслідок цього має максимальну подільність та живучість.

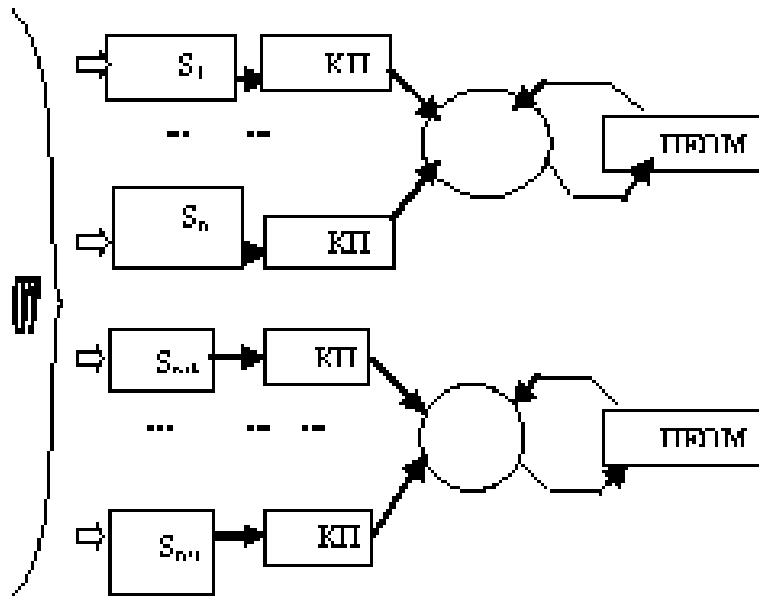


Рис.3.3. Монопольна архітектура

Крім того, кожен оператор володіє всіма ресурсами персональних ЕОМ, включаючи повний об'єм пам'яті, швидкодію, операційне та прикладне програмне забезпечення, час.

Основними недоліками такої архітектури є:

- відсутність інформаційних зв'язків між комп'ютерами, що не дозволяє операторам використовувати масиви даних та прикладні програми інших операторів без фізичного переміщення носіїв даних;
- надзвичайно висока собівартість опрацювання даних, обумовлена невідповідністю ресурсів операторів і ПЕОМ. Наприклад, швидкість формування даних оператором з клавіатури 5–12 операцій/с, а швидкість опрацювання даних ПЕОМ $10^6 - 10^9$ операцій/с;
- недостатнє використання часового ресурсу (4–6 год./добу, що складає 25% потенційних можливостей ПК);
- непрофесійність, недостатня кваліфікація та низька математична підготовка операторів, які не використовують всі можливості операційної системи та прикладного програмного забезпечення. Тому експертна оцінка ефективності монопольної архітектури складає 0,01 % ККД.

3.1.1.2. КСУ розділеного часу

Архітектура розподіленого часу (рис.3.4) не забезпечує високопаралельного режиму роботи, що обумовлено наявністю комутатора інформаційних потоків.

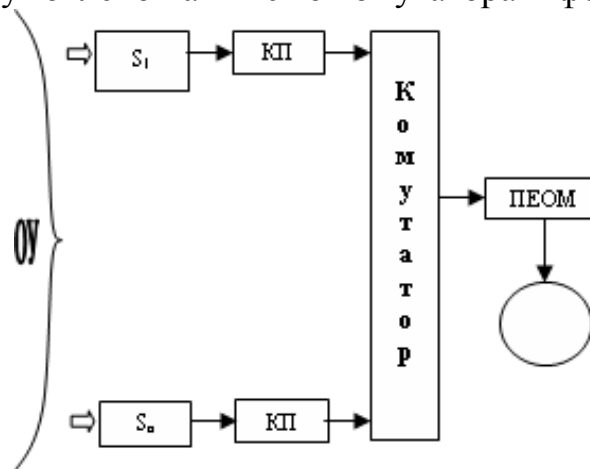


Рис.3.4. Архітектура розподіленого часу

При цьому також виникають ефекти старіння інформації. Позитивними характеристиками такої архітектури є зниження собівартості опрацювання даних за рахунок більш ефективного завантаження КС в часі та колективного користування ПЗ і масивами даних.

Основними недоліками даної архітектури є суттєве зниження надійності, яке обумовлене наявністю комутатора та одного колективного процесора, а також обмеження ресурсів часу для кожного оператора та монопрограмне рішення задач, що може призводити до створення черг на рівні операторів (ККД = 0,1 %).

3.1.1.3. Мультипрограмна КСУ

Мультипрограмна архітектура (рис.3.5) дозволяє розпаралелити інформаційні потоки та організувати одночасне виконання всіх задач в мультипрограмному режимі.

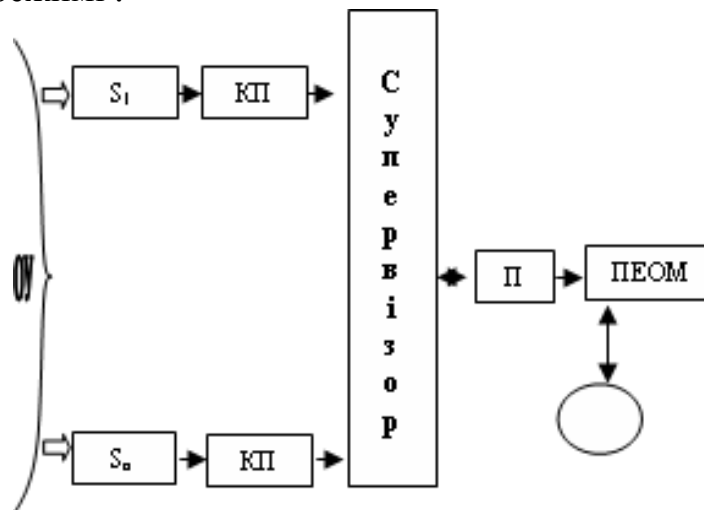


Рис.3.5. Мультипрограмна архітектура.

Недоліками такої архітектури є значна вартість опрацювання даних внаслідок використання одного потужного процесора та складного ПЗ з багаторівневою системою переривань та пріоритетів. Дана структура характеризується невисокою надійністю, великою ймовірністю зависання та невизначеністю часу завершення конкретних задач, що породжує невизначеність часу очікування окремих операторів. Крім того, завантаженість системи залежить від активності операторів (ККД= 1,0–1,2%).

3.1.1.4. Мультипроцесорна КСУ

Мультипроцесорна архітектура (рис.3.6) забезпечує суттєве підвищення надійності системи, можливості розпаралелення інформаційних потоків та суттєве зниження собівартості опрацювання даних за рахунок одночасного використання супервізора та групи процесорів різної потужності.

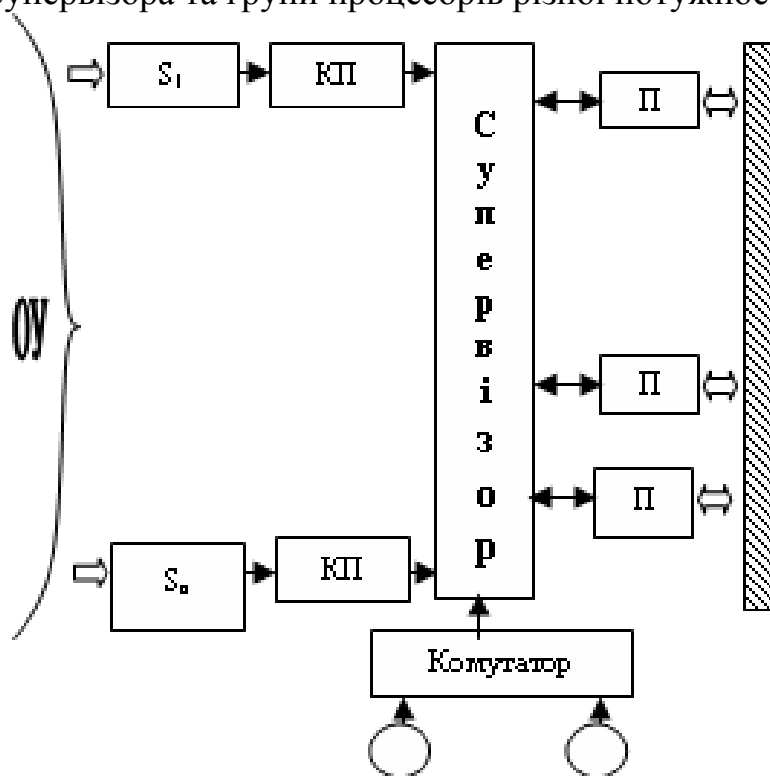


Рис.3.6. Мультипроцесорна архітектура.

При цьому супервізор не тільки аналізує активність та характер задач, які виконуються операторами, але й розподіл потужностей процесорів та їх головних ресурсів. Незалежність від активності сенсорних даних забезпечується цілодобовим рішенням фонових задач, які завантажуються в супервізор операторами через комутатор. Позитивною характеристикою даної архітектури є наявність прямих інформаційних зв'язків між процесорами на основі високошвидкісної паралельної шини (ККД= 5–10%).

3.1.2. Мережеві однорівневі КСУ

3.1.2.1. Магістральна архітектура КСУ.

Магістральна архітектура (рис.3.7) характеризується зниженням собівартості опрацювання розподілених даних за рахунок суттєвого зниження вартості магістрального фізичного каналу зв'язку на основі провідних ліній (витої пари та коаксіального кабеля) або волоконно-оптичних ліній. Головними перевагами такої архітектури є можливість встановлення безпосередніх інформаційних зв'язків між станціями, а також колективне використання ресурсів сенсорних даних та сервера. В той же час дана архітектура має ряд суттєвих недоліків, обумовлених можливістю колізій та конфліктів, які ліквідуються на основі спеціальних складних протоколів доступу, а також низьку надійність, обумовлену одним каналом зв'язку. Незважаючи на ці недоліки, ККД становить 20–30%.

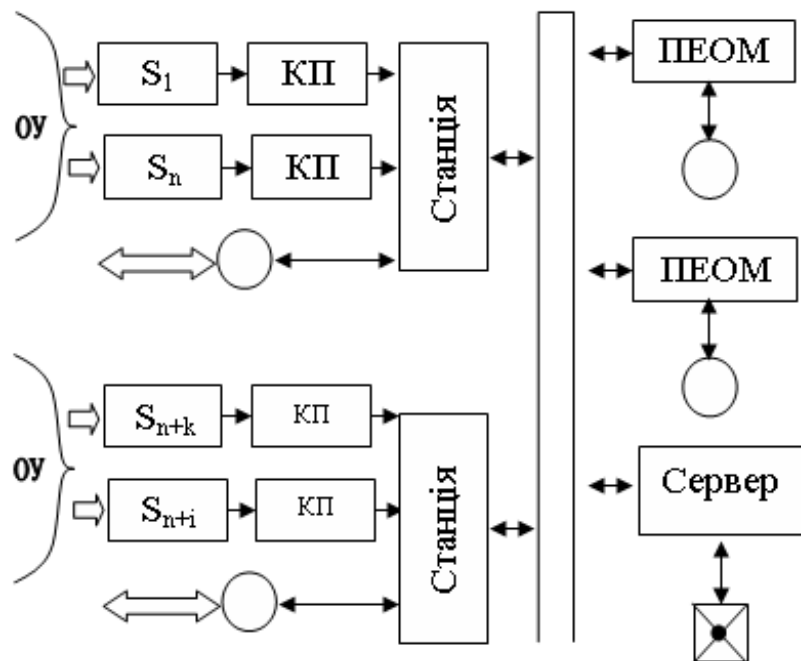


Рис.3.7. Магістральна архітектура.

3.1.2.2. Кільцева архітектура КСУ.

Кільцева архітектура (рис.3.8) найбільш ефективна з даного класу архітектур. Характеризується низькою вартістю каналу зв'язку, що представляє собою шинну магістраль, замкнуту в кільце. Швидкісний маркер, який циркулює в кільці, виключає колізії та спрощує протоколи доступу до каналу. При однократному розриві каналу зв'язку робота мережі не порушується, а по часу поширення сигналів між станціями сервер може локалізувати місце розриву та видати повідомлення про необхідність його профілактики.

Головним недоліком такої архітектури є можливість довготривалої затримки маркера на станції, яка вийшла з ладу, що потребує додаткового тестування мережі з метою вилучення таких станцій.

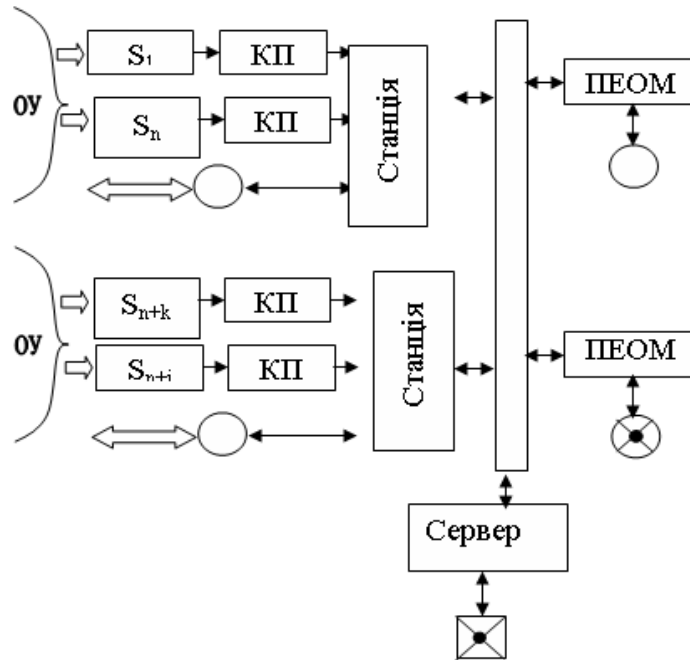


Рис.3.8. Кільцева архітектура.

ККД становить 30–40%. В окремих випадках організовується реалізація каналу на основі подвійного кільця, що суттєво підвищує надійність такої мережі. Тоді ККД може становити 50%.

3.1.2.3. Зіркова архітектура КСУ.

Зіркова архітектура (рис.3.9) найчастіше використовується для створення ЛОМ з концентрованою БД на рівні сервера.

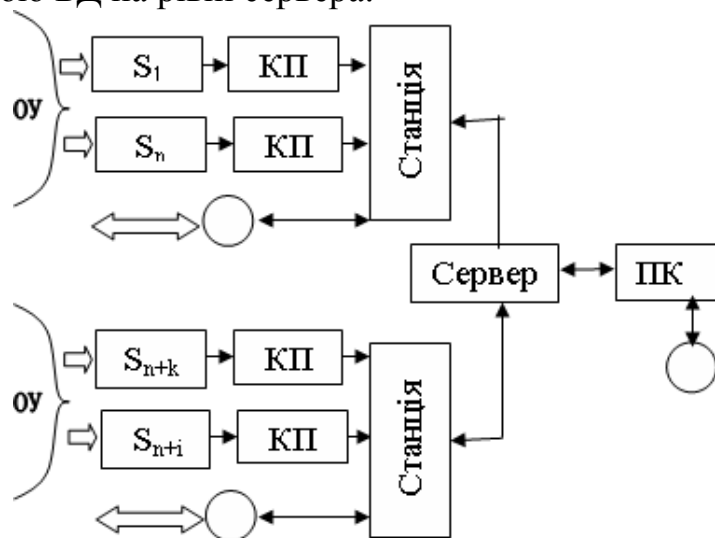


Рис.3.9. Зіркова архітектура.

В даній архітектурі відсутні колізії та організуються прямі інформаційні зв'язки між станціями за рахунок високошвидкісного матричного комутатора. Перевагою даної архітектури є колективне користування ресурсами потужного сервера. Основним недоліком зіркової архітектури є відносно висока вартість системи каналів зв'язків та недостатня надійність, обумовлена комунікаціями через один сервер. В той же час ККД становить 20–30%.

3.1.2.4. Систолічна архітектура КСУ.

Систолічна архітектура (рис.3.10) базується на організації всіх прямих інформаційних зв'язків між будь-якою парою станцій, що забезпечує максимальну надійність телекомунікаційної системи.

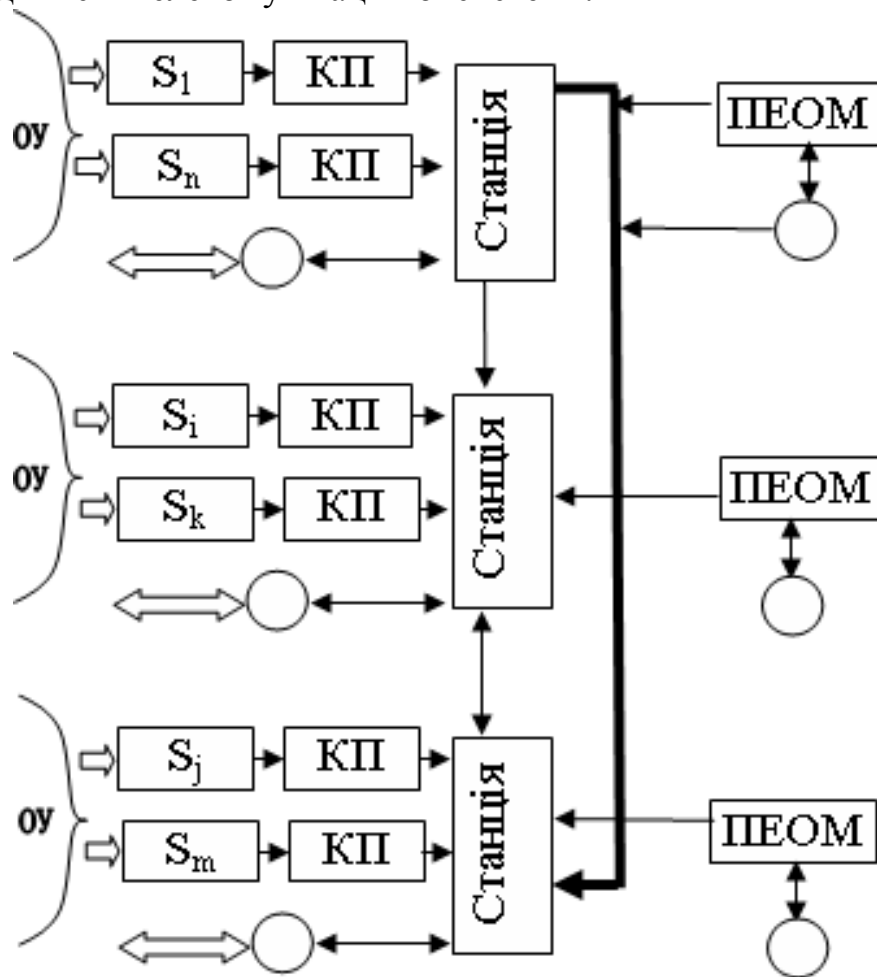


Рис.3.10. Систолічна архітектура.

До класу таких мереж належать СПД системи з комутацією каналів. ККД становить 50–60%, але в той же час ця архітектура характеризується двома основними недоліками: максимальною вартістю каналів зв'язку та можливістю кластерного збудження мережі. Систолічні архітектури є високо емерджентні $K_e > 2$.

Контрольні запитання і завдання.

1. Назвіть класи архітектур КСУ.
2. Класифікуйте архітектури КСУ концентрованої обробки даних.
3. Класифікуйте архітектури КСУ розподіленого опрацювання даних.
4. Охарактеризуйте архітектури КСУ з безпроводними електромагнітними каналами.
5. Систематизуйте архітектури КСУ з оптичними каналами зв'язку.
6. Яка ефективність монопольних КСУ.
7. Порівняйте характеристики КСУ монопольного та розподіленого часу.
8. Вкажіть особливості мультипрограмних та мультипроцесорних КСУ.
9. Якими недоліками характеризуються однорівневі архітектури КСУ?

4. БАГАТОРІВНЕВІ АРХІТЕКТУРИ СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

4.1. Ієрархічна архітектура КСУ.

Ієрархічна архітектура (рис.4.1) розподілених систем опрацювання даних багаторазово повторює зіркову архітектуру, тобто є її розширенням.

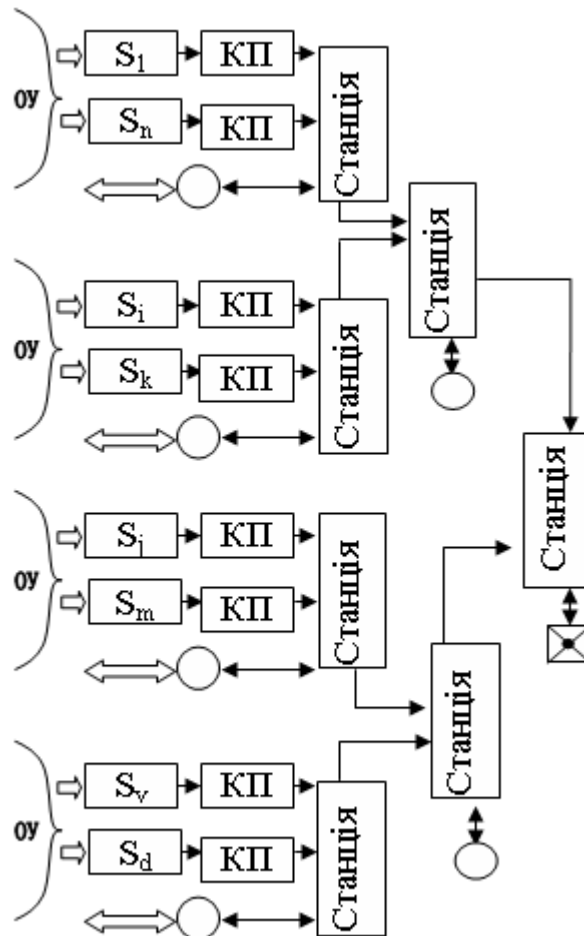


Рис.4.1. Ієрархічна архітектура.

Головною перевагою ієрархічної архітектури є зниження собівартості опрацювання даних за рахунок адаптації ресурсів КМ на кожному рівні або в окремих ієрархічних вітках. Такі архітектури характеризуються максимальною стійкістю. В той же час кожна станція або окремі абоненти мають можливість користуватися потужними обчислювальними ресурсами інших рівнів. ККД таких архітектур становить 40–60%.

Головними недоліками даної архітектури є:

- відсутність прямих інформаційних зв'язків між станціями одного рівня;
- низька емерджентність (інтелектуальність), яка визначається відношенням числа зв'язків до числа станцій, $K_e < 1$;

- можливість катастрофічного розмноження помилок при переміщенні даних з верхніх рівнів до нижніх;
- зростання ваги помилок при переміщенні даних з нижніх рівнів до верхніх.

4. 2. Мережно-ієрархічна архітектура КСУ.

Мережно-ієрархічна архітектура (рис.4.2) одночасно об'єднує елементи багаторівневої мережної та ієрархічної архітектур.

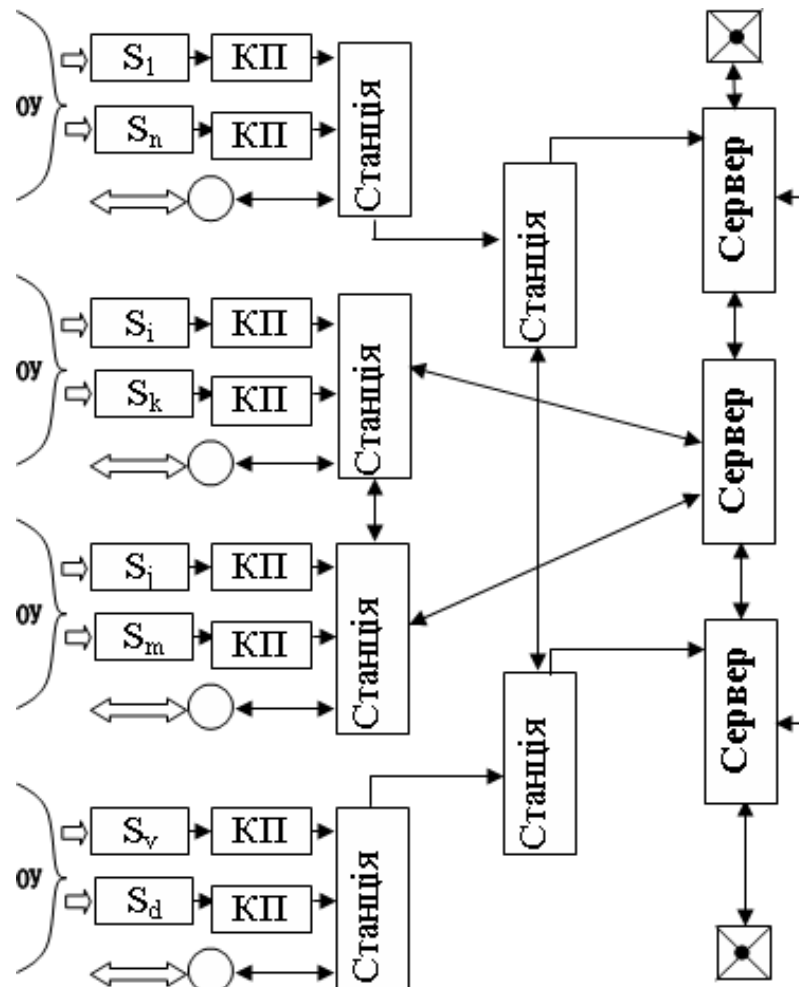


Рис. 4.2. Багаторівнева мережно-ієрархічна архітектура.

Дана архітектура відрізняється наявністю групи рівноправних серверів, об'єднаних швидкісною інформаційною шиною верхнього рівня КСУ. Даний елемент архітектури забезпечує високий інтелект та надійність системи. Особливістю такої архітектури є наявність зв'язків між окремими станціями на певних рівнях, а також через рівневі зв'язки між серверами верхнього рівня та будь-якого іншого рівня ієрархічної компоненти архітектури КСУ. При цьому у процесі проектування КСУ реалізується максимальна адаптація архітектури КСУ до топології реального розподіленого ОУ.

Одним з недоліків такої архітектури КСУ є складність організації та ведення розподіленої бази даних на рівні серверів.

4.3. Багаторівнева магістральна архітектура КСУ.

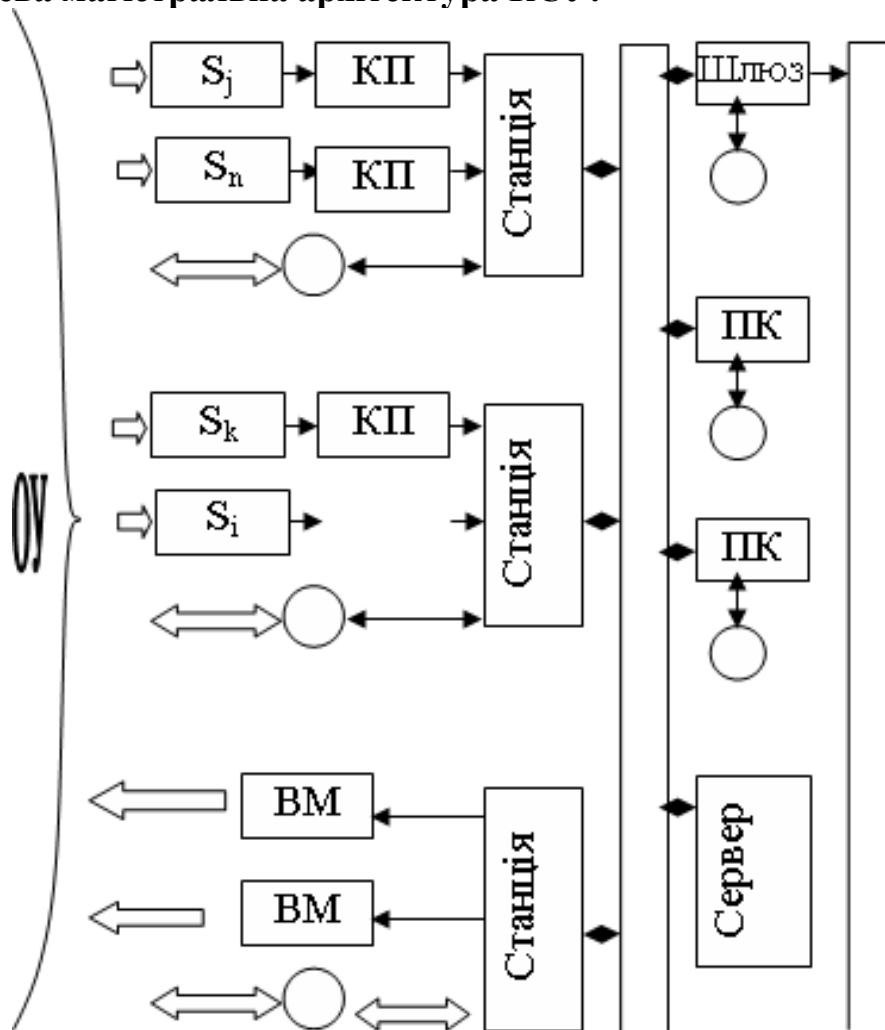


Рис. 4.3. Багаторівнева магістральна архітектура.

Дана архітектура (рис.4.3) максимально адаптована до структури промислових виробництв і охоплює три рівні: технологічний (сенсори, виконавчі механізми, оператори, технологи та станції, кодери), цеховий (ЕОМ типу сервер та цехові ПК), адміністративний (ПЕОМ адміністративного апарату та системний сервер з адміністратором БД). ККД становить 50–65%.

4.4. Зірково-магістральна архітектура КСУ.

Дана архітектура (рис.4.4) характеризується суттєвими перевагами по відношенню до інших описаних архітектур, що виражається в наступному:

- багаторівнева розподілена обробка даних;

- відсутність міжрівневих шлюзів, які значно обмежують ресурси багаторівневих систем;
- висока надійність і живучість за рахунок реалізації розпаралелення обробки даних, в тому числі при черезрівневих обмінах;
- висока емерджентність;
- реалізація принципів паралелізму на рівні сенсорів та ПКД.

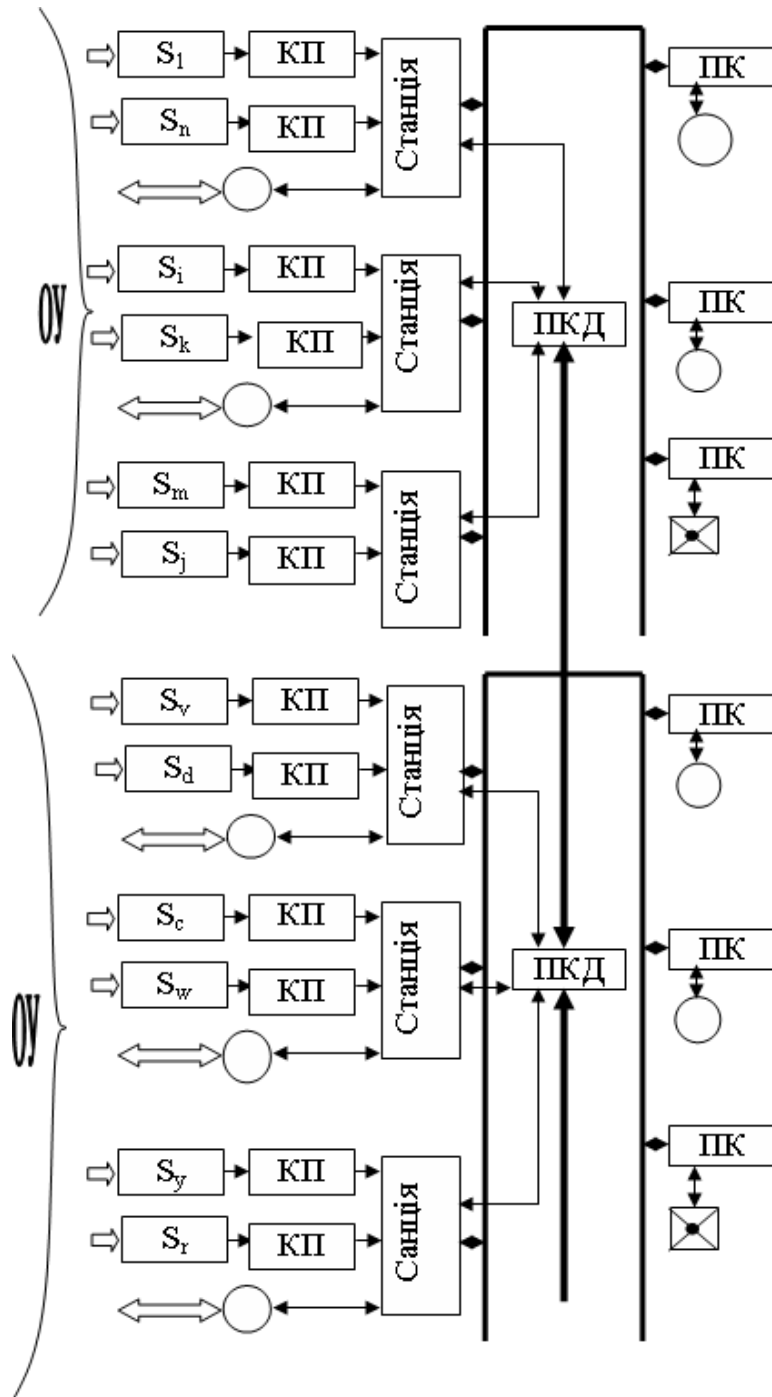


Рис. 4.4. Зірково-магістральна архітектура.

Особливістю даної архітектури є організація паралельного запису даних в поштові скриньки асоціативних елементів ПКД та можливість паралельного зчитування всіма абонентами будь-яких даних всіх елементів ПКД.

Швидкісний канал міжрівневого зв'язку між ПКД, як правило, реалізується на основі волоконно-оптичних ліній зв'язку, що дозволяє вирівняти трафіки магістральних ліній кожного рівня та розпаралелити міжрівневі зв'язки.

Оцінка ККД зірково-магістральної архітектури сягає 60–80%. Тобто можна вважати дані архітектури найбільш перспективними в застосуванні у КС збору, обробки даних та управління складними об'єктами в реальному масштабі часу.

4.5. Поняття, архітектура та функції мережевої станції КСУ

Мережеві архітектури КСУ організуються на основі реалізації функцій формування, цифрового опрацювання, зберігання, управління об'єктами та передавання інформації мережевими станціями (СТ) з базовою архітектурою, яка показана на рис. 4.5. На рис. 4.5: ОУ - об'єкт управління; О - оператор; АП - абонентський пункт; Р - процесор; ММ - модем; КМ - комп'ютерна мережа; S - сенсор; КНМ - контролер низової мережі; ВМ - виконавчий механізм.

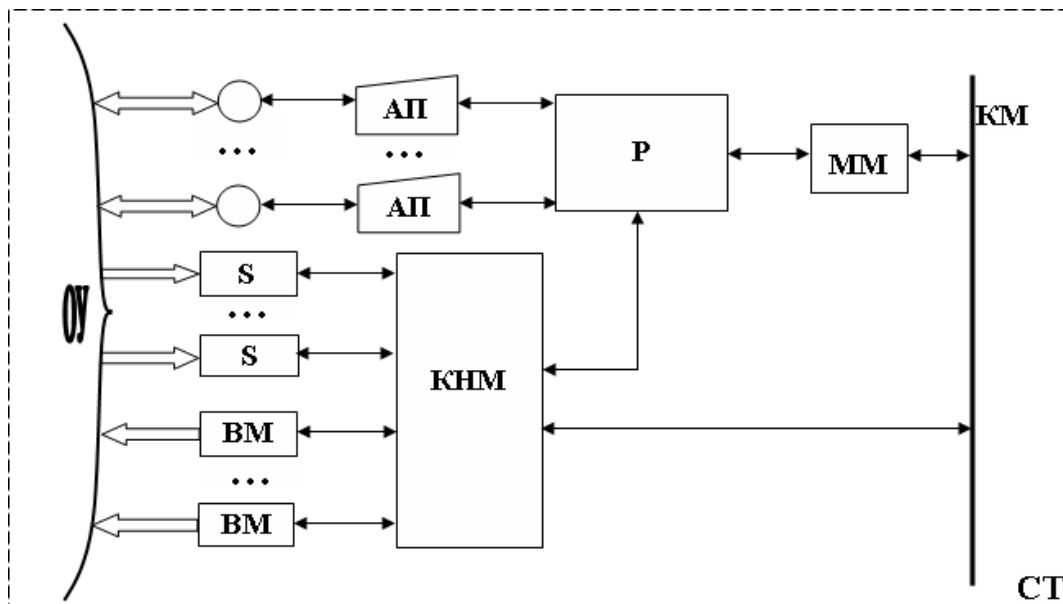


Рис. 4.5. Архітектура мережевої станції КСУ.

Компоненти мережевої станції виконують наступні функції:

- збір та аналіз технологічних та техніко-економічних даних (ТЕД) в реальному масштабі часу виконують оператори, оснащені спеціалізованими абонентськими пунктами у вигляді промислових комп'ютерів, спеціалізованих пультів, табло, панелей перемикачів та ін.;

- сенсори реалізують перетворення фізичних процесів в стандартні сигнали та інтерфейси;

- виконавчі механізми реалізують команди управління механізмами, автоматичними засувками, роботами, верстатами з програмним управлінням...;

- контролер низової мережі реалізується на основі двопроцесорної структури у вигляді спеціалізованого мікроконтролера з оптоелектронною гальванічною розв'язкою пиле-, волого- та вибухобезпечного виконання, в якому один процесор виконує функції збору, кодування, реєстрації технологічних даних та управління мікрокомандами виконавчих механізмів, а другий процесор реалізує функції телекомунікаційного зв'язку з базовим процесором станції, а в окремих випадках

– згідно інтерфейсу низової комп'ютерної мережі ресурсами влаштованого модему;

- базовий процесор станції Р виконує функції оперативної архівації даних на заданому інтервалі часу, побудові інформаційних моделей ОУ згідно характеристичного функціоналу X_{OU} , контролю відхилень станів ОУ від норми та організації мережевого зв'язку через модем з іншими станціями та серверами КСУ.

Контрольні запитання і завдання.

1. Назвіть компоненти багаторівневих магістральних КСУ.
2. Поясніть характеристики пам'яті колективного доступу зірково-магістральних архітектур КСУ.
3. Назвіть класи та характеристики архітектур безпроводних КСУ на основі радіоканалів.
4. Класифікуйте КСУ з відкритими оптичними каналами.
5. Поясніть властивість емерджентності мережевих архітектур КСУ.

5. ЗАКОНИ ДОЦІЛЬНОСТІ ПРОЕКТНИХ РІШЕНЬ КСУ

У загальному випадку під законами економічної доцільності розуміють ряд критеріїв доцільності, які описуються системою характеристичних функцій та дозволяють оптимізувати РКС на основі законів: фрактальності; Гроша; тах прибутку; якості; собівартості.

5.1. Закон Фрактальності.

Закон фрактальності, застосований до архітектури комп'ютерних мереж, встановлює співвідношення між мережею і підмережею, мережею і фрагментом мережі і описується рівнянням

$$\hat{O} = \frac{\hat{O}_{i+1}}{\hat{O}_i} = const, \quad (5.1)$$

де Φ_{i+1} , Φ_i – відповідно – системні характеристики на границі двох рівнів комп'ютерної мережі.

На рис.5.1 показаний приклад реалізації закону фрактальності в комп'ютерній мережі, де 1 – глобальна мережа; 2 – локальна мережа; 3 – низова комп'ютерна мережа; 4 – користувач.

Виконання закону фрактальності передбачає не тільки ієрархію архітектури комп'ютерної мережі, але й можливість безпосереднього зв'язку користувача з глобальною мережею.

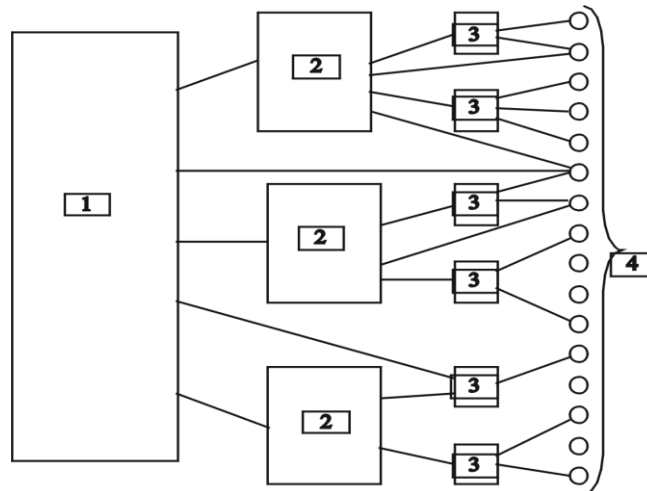


Рис.5.1. Характеристика закону фрактальності при проектуванні комп'ютерних систем.

5.2. Закон Гроша.

Закон Гроша встановлює співвідношення між системними характеристиками проектованої комп'ютерної мережі та системними характеристиками діючих мереж і виражається умовою:

$$G_{i+1}(T > T_0) \geq 2G_i(T > T_0), \quad (5.2)$$

де G_{i+1} G_i – відповідно – системні характеристики проекрованої та діючих комп'ютерних мереж.

На рис.5.3 показаний приклад реалізації закону Гроша, де T_0 – момент часу зміни системних характеристик комп'ютерної мережі; $B(T)$ – епіюра економічних затрат, необхідних для покращення системних характеристик комп'ютерної системи.

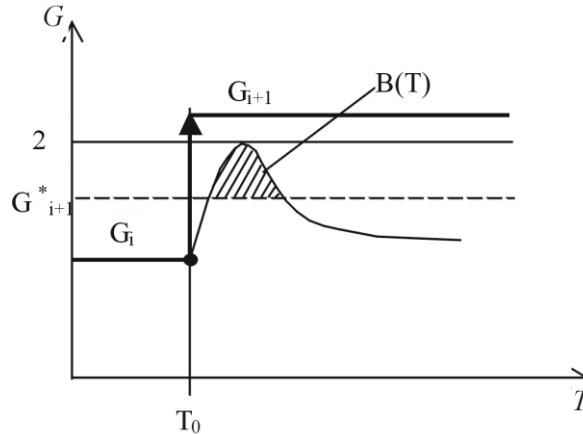


Рис.5.3. Характеристика закону Гроша

З рис.5.3 видно, що якщо G_{i+1} не задовільняє вираз (5.2), що показано пунктиром, G_{i+1}^* , то економічні затрати будуть перевищувати прибуток, і така зміна системних характеристик є економічно не доцільною і може привести до деградації комп'ютерної мережі та банкрутства фірми-виробника.

5.3. Закон максимального прибутку.

Закон max прибутку характеризує часові характеристики та конкуренцію впровадження проекрованої комп'ютерної мережі (рис.5.4) і виражається рівняннями:

$$P(T_1)_{\max} > P(T_{i+1})_{\max} ; \quad (5.3)$$

$$P(T_i)_{\max} = \int_{T_0}^{\infty} P(T) dT . \quad (5.4)$$

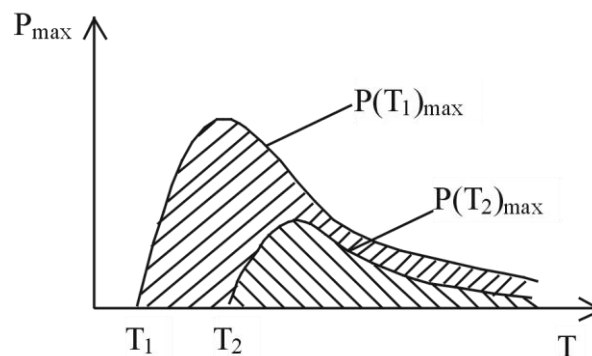


Рис.5.4. Характеристика закону max прибутку при однаковій потужності виробників нових засобів комп'ютерних мереж.

Дана характеристика закону тах прибутку відповідає умовам обмеженої, приблизно однакової, потужності проектних організацій та організаторів масового випуску комп'ютерних мереж.

Зниження прибутків пропорційно часовому параметру відповідно моральному старінню комп'ютерного обладнання та ін.

При різній потужності проектних організацій та відповідних підприємств тиражування закон тах прибутку має інший характер, який представлений на рис.5.5 та описується системою рівнянь (5.5):

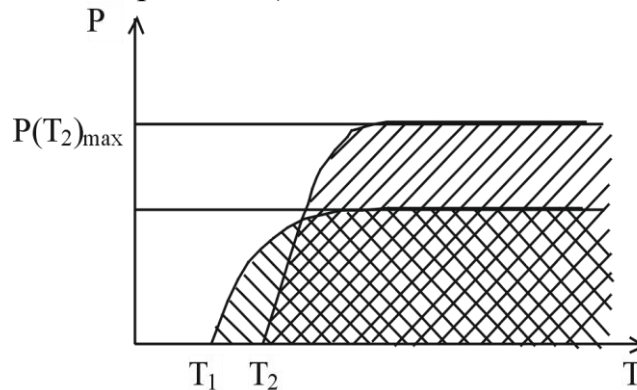


Рис.5.5. Характеристика закону тах прибутку при різній потужності проєктантів комп'ютерних мереж.

$$\left\{ \begin{array}{l} P(T_1)_{\max} < P(T_2)_{\max} = \text{const}; \\ P(T_1)_{\max} = \int_{T_1}^{\infty} P(T_1) dT; \\ P(T_2)_{\max} = \int_{T_2}^{\infty} P(T_2) dT - \int_{T_2}^{\infty} P(T_1) dT. \end{array} \right. \quad (5.5)$$

5.4. Закон якості.

Закон якості характеризує системні характеристики комп'ютерних мереж на етапах їх розробки, освоєння виробництва, гарантійної експлуатації та післягарантійного використання. Даний закон описується системою рівнянь (5.6), представлений на рис.5.6 і відповідає розподілу Вейбула, який описує характеристики надійності комп'ютерної системи.

$$\left\{ \begin{array}{l} Y(T_1) - Y^* = \min; \\ Y(T_1) > Y^*; \\ Y(T_1) - Y(T_2) = \min; \\ T(T > T_2) = \text{var}. \end{array} \right. \quad (5.6)$$

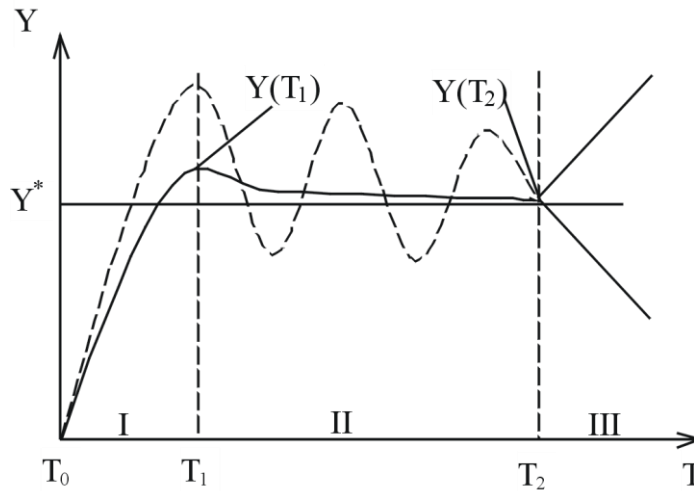


Рис.5.6. Характеристика закону якості.

На рис.5.6: I, II, III – відповідно функції якості на етапах проектування, гарантійної та післягарантійної експлуатації комп’ютерної мережі. Пунктиром показано характеристики підтримки якості при ремонтах та відновленні гарантійних функцій комп’ютерної мережі.

Суть закону якості полягає в тому, що на етапі проектування, освоєння виробництва нової комп’ютерної мережі, її системні характеристики повинні перевищувати світовий рівень якості на мінімальну величину, при цьому повинно забезпечуватися мінімальне перевитрат виробництва, а на етапі гарантійної експлуатації забезпечуватися мінімальне зниження системних характеристик комп’ютерної мережі. В післягарантійний період характеристики мережі не регламентуються.

5.5. Закон собівартості.

Закон собівартості визначає характер зміни собівартості створення та впровадження нових комп’ютерних мереж в залежності від умов зміни їх системних характеристик. В першому варіанті задається умова підтримки заданих системних характеристик комп’ютерної мережі при нарощуванні її виробництва та тиражуванні. Даний варіант закону собівартості описується системою рівнянь:

$$\begin{cases} S(T) = \text{const}; \\ N(T) = \text{var}; \\ P(T) = (Z(T) - C(T)) * N(T). \end{cases} \quad (5.7)$$

На рис.5.7 показаний приклад реалізації закону собівартості.

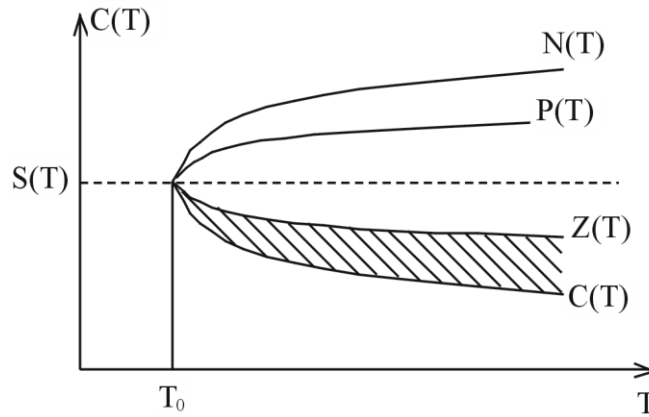


Рис.5.7. Характеристика закону собівартості (перша модифікація).
 $S(T)$ – системні характеристики мережі; $N(T)$ – об’єм виробництва; $P(T)$ – прибутки; $Z(T)$ – вартість мережі; $C(T)$ – собівартість виробництва.

Собівартість виробництва визначається згідно виразу:

$$C_T = \sum_{i=1}^m C_i N_i ,$$

де m - кількість видів продукції підприємства;

C_i - повна собівартість i -го виду продукції підприємства;

N_i - обсяг випуску продукції i -го виду в натуральному вираженні.

Другий варіант закону собівартості (рис.5.8) визначається на основі проєктованого покращення системних характеристик створюваних комп’ютерних мереж при незмінному об’ємі випуску і описується рівняннями:

$$\begin{cases} N(T) = const; \\ S(T) = var; \\ P(T) = (Z(T) - C(T)) * N(T). \end{cases} \quad (5.8)$$

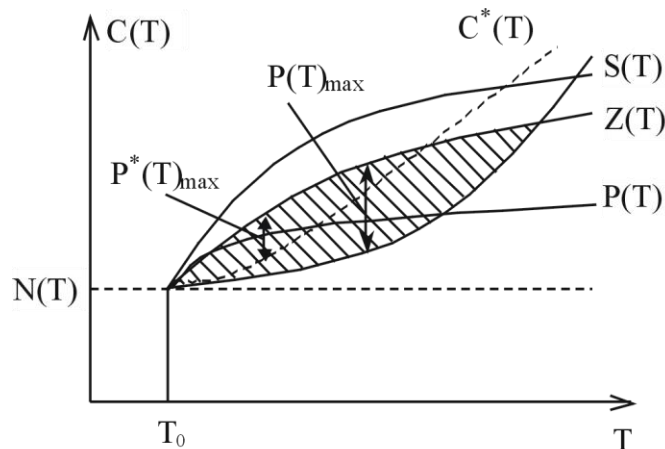


Рис.5.8. Демонстрація закону собівартості (друга модифікація).

З рис. 5.8 видно, що \max прибуток виникає при \max різниці між вартістю продажі і собівартістю виробництва комп'ютерної мережі. Очевидно, що при відсутності досконалої технології виробництва відмічається суттєве зростання собівартості виробництва $C(T)$, що показано на рисунку пунктиром. При цьому \max прибутки $P^*(T)$ набагато менші від прибутків, досягнутих за рахунок досконалої технології, тобто $P^*(T)\max \ll P(T)\max$. Крім того, виробництво з низьким рівнем технології, в принципі, не може випускати комп'ютерні мережі з високим рівнем системних характеристик, інакше воно буде збитковим.

Слід зауважити, що впровадження нових технологій виробництва за рахунок винаходів та “ноу-хау” на рис.5.8 штрих-пунктир $C^0(T)$, то прибутки $P^0(T)$ будуть набагато більші по відношенню до $P(T)\max$, і комп'ютерна мережа буде характеризуватися досконалими системними характеристиками.

Таким чином, викладені закони доцільності розробки та впровадження комп'ютерних мереж з покращеними системними характеристиками демонструють основні напрямки та базові можливості зростання їх економічної доцільності. При проектуванні комп'ютерних мереж та їх компонентів, особливо це стосується засобів формування ТЕД на основі викладених законів економічної доцільності, може бути ефективно використана технологія побудови матричних моделей руху даних, а також розроблена інформаційна технологія економічних епюр затрат і прибутків для кожного елемента матричних моделей.

Контрольні запитання і завдання.

1. Охарактеризуйте етапи проектування КСУ в залежності від готовності, вартості, економічного ефекту, впровадження та затрат часу на реалізацію функцій у вузлах КСУ.
2. Назвіть характеристики закону фрактальності та Гроша.
3. Якими характеристиками описується закон \max прибутку?
4. Охарактеризуйте етапи та функції закону якості.
5. Опишіть модифікації закону собівартості.

6. МЕРЕЖІ ПЕТРІ ТА ЇХ ЗАСТОСУВАННЯ ПРИ ПРОЕКТУВАННІ СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

Як зазначалося, об'єктом досліджень даного предмету є мережні системи. Для формалізації структури таких систем застосовується теорія графів і мереж з наступними визначеннями :

Графом називається сукупність двох множин: множини точок, що називаються вершинами або вузлами графа, і множини упорядкованих пар цих вершин, які називаються дугами або ребрами. Коли порядок вершин не має значення, зв'язок між вершинами називається ребром. Якщо порядок вершин строго визначений, зв'язок називається дугою.

Приклад графа наведено на рис.6.1. Вершини графа позначено латинськими буквами a, b, c, d, e , дуги – грецькими буквами – $\alpha, \beta, \delta, \varphi, \varepsilon, \gamma, \zeta$.

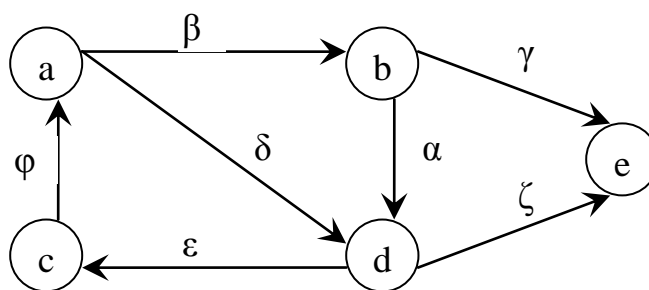


Рис. 6.1 Приклад графа

Це далеко не єдиний спосіб позначення вершин. Вершини можуть також позначатися цифрами. Іноді можливе описове позначення вершин. В останньому випадку в кружечки, що позначають вершини, вписують будь-які символи, зрозумілі для того, хто вирішує конкретну проблему мовою теорії графів.

Граф, у якому зв'язки між вершинами задаються ребрами, називаються неорієнтованим.

Граф, у якому зв'язки між вершинами задаються дугами, називаються орієнтованим.

На рис. 6.1 зображено орієнтований граф. Для того, щоб він став неорієнтованим, треба на дугах забрати стрілки.

Ребра і дуги зручно позначати переліченням пар вершин, які вони з'єднують. На першому місці ставиться початкова вершина. На другому – кінцева. У цьому випадку дуга φ позначається як (c, a) , дуга β як (a, b) і т.д. Для ребер порядок вершин не має значення.

Довільний граф позначається буквою G , X – множина вершин, E – множина ребер, A – множина дуг.

Тоді конкретний граф, залежно від його орієнтації, прийнято записувати у вигляді:

$$G = (X, A) \text{ або } G = G(X, A), G = (X, E) \text{ або } G = G(X, E).$$

Початкова і кінцева вершини дуги чи ребра називаються кінцевими. На рис. 6.1, наприклад, для дуги β кінцевими є вершини a і b .

Вершина називається інцидентною дузі чи ребру, якщо вона є для нього початком або кінцем. На тому ж рисунку, наприклад, вершина a інцидент на дугам φ , β і δ .

Дуги інцидентні одна одній, якщо вони інцидентні одній і тій самій вершині (тобто мають у своєму складі ту саму вершину). Наприклад, φ , β , δ інцидентні вершині a .

Дуга початкова і кінцева, вершини якої збігаються, називаються петлею. Вигляд її подано на рис. 6.2.

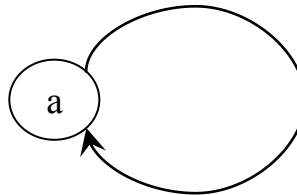


Рис. 6.2 Петля.

У реальних задачах петля означає „зациклення” або „топтання” на одному місці: наприклад, ресурс чи інформація не виходить із джерела і залишається в ньому. Це досить поширене явище на практиці.

Дві вершини називаються сусідніми, якщо є ребро чи дуга, що їх з'єднують.

Довільна послідовність дуг або ребер називається ланцюгом. Для ланцюга важливо, що кожне наступне ребро або дуга були початком чи кінцем попереднього ребра або дуги. Однак, спрямованість дуг у ланцюзі не має значення, тобто в ланцюзі допускаються повернення. Наприклад, на рис. 3.3 вершини a і e з'єднані ланцюгами: α , β , γ і ε або α , β і δ . І це не всі ланцюги між a і e на цьому рисунку.

Ланцюг, у якого початкова і кінцева вершини збігаються, називаються циклом. На рис. 6.3. прикладом циклу є дуги γ , ψ і δ .

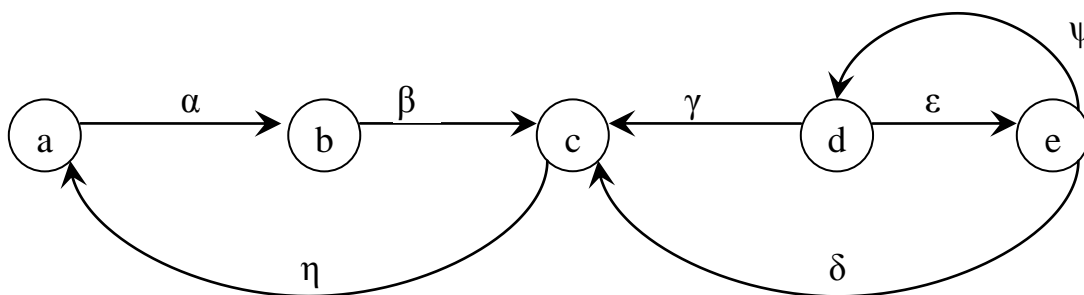


Рис. 6.3 Приклад графа.

Ланцюг, у якого повернення між сусідніми вершинами не допускаються (тобто такий ланцюг, що складається з дуг одного напрямку), називається шляхом. Шляху між вершинами a й e на рис. 6.3 немає.

Шлях, у якого початкова і кінцева вершини збігаються, називається контуром. На рис. 6.3 шлях утворюють дуги α і β . Дуги α , β і η утворюють контур.

Для неорієнтованого графа поняття ланцюга і шляху, а також циклу і контуру збігаються.

Граф називається зв'язним, якщо в ньому для кожної пари вершин знайдеться ланцюг, що їх з'єднає. Будь-який граф можна розглядати як деяку сукупність зв'язних графів. Кожен з таких графів називається під графом або компонентом вихідного графа.

Усі дотепер розглянуті графи були однозв'язними. На рис.6.4 подано двозв'язний граф, що складається з двох компонентів. Перший з них визначається дугами α і β , другий – дугами δ , ε і ξ .

Сукупність дуг графа називається деревом, якщо вона є зв'язним під графом цього графа і не містить циклів. На рис.6.4 вказано граф, що складається з двох під графів. Обидва підграфи є деревом.

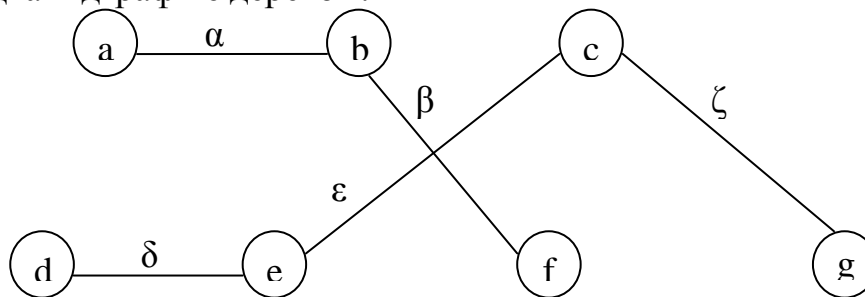


Рис.6.4 Приклад багатозв'язного графа

Щоб компоненти розглянутого графа подати наочно, можна зобразити їх окремо, як на рис. 6.5. Збереження масштабу тут принципового значення не має.

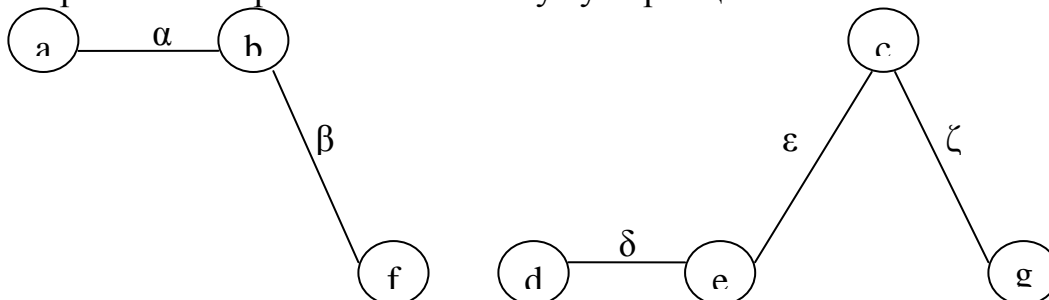


Рис. 6.5 Багатозв'язний граф для графа на рис. 2.4, зображений у вигляді окремих дерев.

Вагою дерева називається сума ваг усіх його ребер. Дерево називається мінімальним (максимальним), якщо воно має мінімальну (максимальну) з можливих ваг.

Лісом називається будь-яка сукупність дуг, що не містить циклів. Ліс може складатися з одного і більше дерев. На рис. 6.5 зображено ліс, що складається з двох дерев: дерева α , β і дерева δ , ε , ξ .

Вага лісу вводиться за аналогією з вагою дерева.

Покривним деревом називається будь-яке дерево, що містить усі вершини графа. Зрозуміло, що для будь-якого зв'язного неорієнтованого графа покривне дерево завжди існує. Для зв'язного орієнтованого – далеко не завжди.

Приклад цього положення ілюструє рис.6.6. Для орієнтованого графа на цьому рисунку не існує покривного дерева, хоча за відсутності стрілок на дугах воно склалося б з ребер (1, 2), (2, 3) і (3, 4).

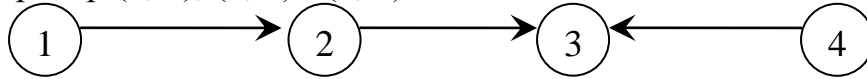


Рис. 6.6 Приклад графа, для якого не існує покривного дерева.

Тому для орієнтованих графів на перше місце виступає поняття лісу, а не дерева. Для того, щоб чітко визначитися в термінах, введемо їх окремо для орієнтованих графів.

Орієнтоване дерево – дерево, у якому жодні 2 дуги не заходять у ту саму вершину.

Коренем дерева називається єдина його вершина, в яку не заходить жодна дуга. Поняття кореня для неорієнтованих дерев немає. Коренем для орієнтованих дерев може слугувати будь-яка вершина.

Орієнтований ліс – це ліс, що складається з орієнтованих дерев.

Покривним орієнтованим деревом називається орієнтоване дерево, що є одночасно і покривним деревом.

Покривним орієнтованим лісом називається орієнтований ліс, що містить всі вершини графа.

На рис.6.6, як вже зазначалося, показаний граф, що не має покривного дерева. Покривний же ліс для нього складається з двох дерев: дерева (1, 2), (2, 3) і дерева (4, 3). Останнє дерево містить тільки одну дугу, але це не заважає йому бути, відповідно до вищенаведеного визначення, деревом.

Граф, що має покривне дерево, є однозв'язним. Вірно і протилежне: для того, щоб граф був однозв'язним, він повинен мати покривне дерево.

У загальному випадку граф може мати кілька покривних дерев. Особливо показове це явище для неорієнтованих графів. Хоча можливе і для орієнтованих. Для того щоб це побачити, розглянемо граф на рис.6.7.

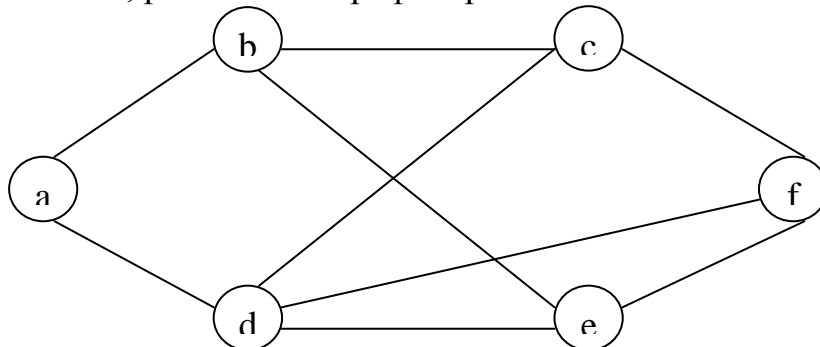


Рис. 6.7 Приклад графа, для якого існує кілька покривних дерев

Для графа на рис.6.7 покривними будуть і дерево $(a, b), (b, c), (c, f), (f, e), (e, d)$ і дерево $(a, b), (b, c), (c, d), (d, f), (f, e)$.

За бажання тут можна знайти й інші покривні дерева. Пропонуємо перевірити це.

Кожне дерево можна представити (розгорнути) таким чином, щоб воно мало вигляд відомого в історичному плані „генеалогічного” дерева, за допомогою якого зображають наявність родинних зв'язків. Вигляд генеалогічного дерева зображено на рис.6.8.

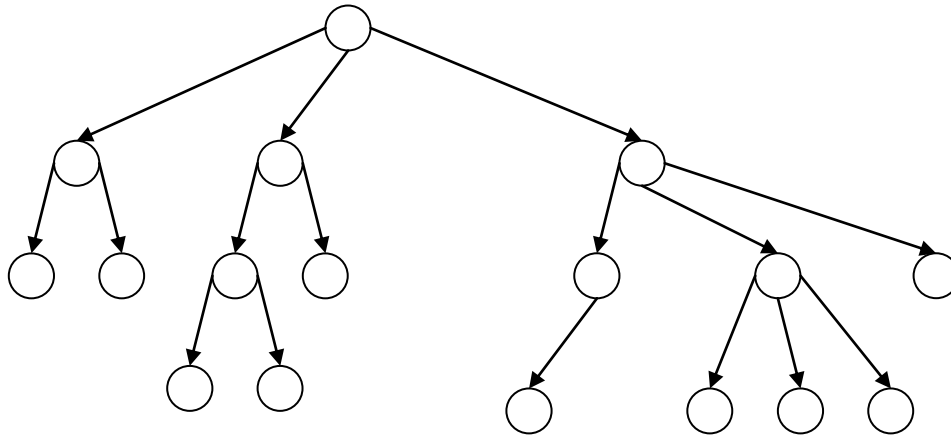


Рис. 6.8 Вигляд генеалогічного дерева.

Так само відбивають різного роду ієрархію і супідрядність для широкого кола реальних об'єктів. При цьому для неорієнтованого дерева як корінь можна вибрати будь-яку вершину, для орієнтованого – одну, а саме ту, в котру не заходять дуги.

Множина дуг, виключення яких із графа збільшує число його компонентів називається розрізом. Розріз, що не містить як власну підмножину ніякого іншого розрізу, називається простим. Таким чином, простий розріз це той, котрий містить найменшу кількість дуг. Наприклад, для графа на рис.6.3 дуги $\gamma, \varepsilon, \psi$ і δ утворять розріз. До складу його входить інший розріз - γ, ε і δ , що є простим.

Узагальненням поняття графа є поняття мережі. Воно вводиться відповідно до поняття мережної системи на загальноприйнятому рівні.

Мережею називається граф, кожному ребру або дузі якого поставлено у відповідність деяке число, що називається вагою ребра, або дуги і відбиває певні властивості цього ребра чи дуги (наприклад, відстань, швидкість, ціну, пропускну здатність, дохід, прибуток).

Мережі на практиці, природно, використовуються набагато частіше, ніж графи як такі, тому що ваги ребер чи дуг визначають процес функціонування об'єкта, представленого мережею. У деяких літературних джерелах поняття мережі і графа не розрізняються.

Усі поняття, уведені для графа, стосуються і мереж. Мережі, як і графи, можуть бути орієнтованими або ні. Вигляд орієнтованої мережі зображено на рис. 6.9. На ньому поруч з кожною дугою проставлено її вагу.

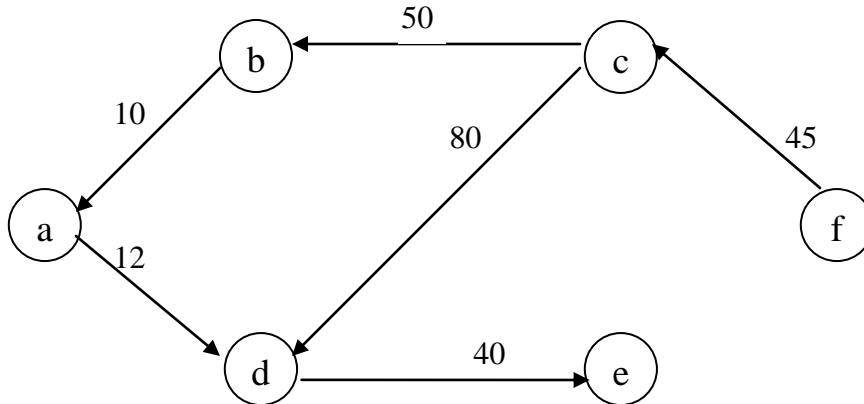


Рис.6.9 Зображення мережі.

Завдання ваг у такий спосіб є не завжди зручним для мереж великого розміру чи при зображенні їх у малому масштабі. Тому часто ваги для мереж задаються у вигляді таблиці. Зразок табличного задання ваг для мережі, зображеної на рис.6.9, наведено у таблиці 6.1.

Таблиця 6.1

Спосіб задання ваг для мережі на рис.2.9.

$y \backslash z$	a	b	c	d	e	f
a		10				
b			50			
c						45
d	12		80		40	
e						
f						

Кажучи загалом, таблиця 6.1 являє собою задання самої мережі. Вона відображає усе, що притаманне мережі: вершини, зв'язки і ваги, тобто числа, поставлені у відповідність цим зв'язкам. За нею, не знаючи конфігурації мережі можна відтворити всю мережу. Масштаб і вигляд зображення мережі для формалізованої задачі не мають принципового значення, крім виконання ролі наочності. Саме так описуються мережі при розв'язанні мережних задач на комп'ютерах. Цей метод зручний для програмування на алгоритмічних мовах.

Взагалі не тільки мережі, а й графи можуть задаватися таблицею. У цьому випадку замість конкретних ваг у відповідних клітинках таблиці ставиться яка-небудь ознака, що вказує на наявність зв'язку.

Матриці інцидентій, які викладені в цьому розділі принципи формалізації опису графів, є теоретичною та методологічною основою для побудови матричних

моделей руху даних. При цьому ММРД об'єднують характеристики матриці інциденцій та графових дерев і більш конкретизовані, оскільки доповнюються символами атрибутів, які відображають поняття джерела інформації, проміжного пункту цифрової обробки даних та приймача інформації.

Контрольні запитання і завдання.

1. Дайте визначення графа.
2. Дайте визначення орієнтованого графа.
3. Дайте визначення неорієнтованого графа.
4. Поясніть принцип побудови ланцюга в теорії графів.
5. Поясніть принцип побудови шляху в теорії графів.
6. Наведіть приклад багатозв'язного графа.

7. МАТРИЧНІ МОДЕЛІ РУХУ ДАНИХ В СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМАХ

7.1. Атрибути матричної моделі.

Матриці інциденцій є теоретичною та методологічною основою для побудови матричних моделей руху даних. При цьому ММРД об'єднують характеристики матриці інциденцій та графових дерев і більш конкретизовані, оскільки доповнюються символами атрибутів, які відображають поняття джерела інформації, проміжного пункту цифрової обробки даних та приймача інформації.

Матрична модель руху даних (рис.7.1) визначається умовами та графом взаємодії об'єктів O_1, \dots, O_4 та документами D_1, \dots, D_4 .

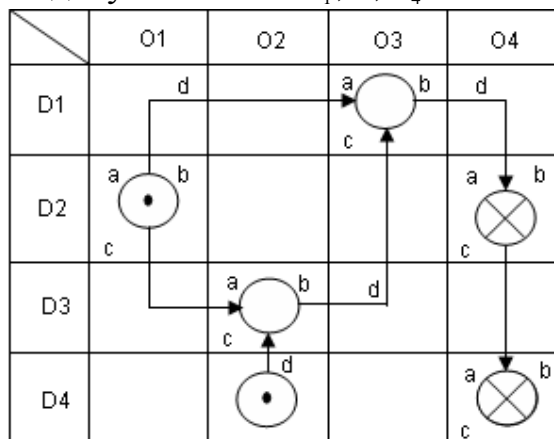


Рис.7.1. Матрична модель руху даних.

При цьому для кожного SO з координатами $D_i \cdot O_j$ повинні виконуватися умови несуперечливості:

- джерело інформації має не менше одного виходу і жодного входу ;
- пункт обробки даних має не менше одного входу і одного виходу ;
- залежний пункт затвердження даних має не менше одного входу і тільки один вихід ;
- незалежний пункт затвердження даних має не менше одного входу і жодного виходу (табл.7.1).

Для формальної побудови сімейства моделей руху даних на основі відомої двомірної ММ кожний елемент двомірної ММ описується четвіркою параметрів: a – початок виконання операції; b – тривалість виконання операції; c – тип операції; d – час передавання даних у каналі зв'язку між об'єктами.



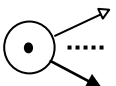
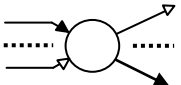
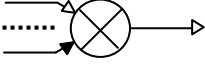
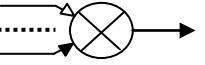
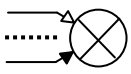
Значний вклад в розвиток теорії проектування комп'ютерних мереж та автоматизованих систем вніс відомий американський вчений Дж. Мартін, який визначив поняття і ввів оцінку одиниці руху даних у вигляді :

$$\hat{E}_d = \frac{R}{W}, \tag{7.1}$$

де R – число зчитувань або запитів,
 W – число записів або оновлень даних.

Таблиця 7.1

Символіка атрибутів ММРД

Символи атрибутів ММРД	Зміст атрибутів ММРД
	інформаційний потік
	матеріальний потік
	джерело
	пункт обробки ІМП (інформаційно-матеріальний потік)
	залежний пункт затвердження
	залежний пункт затвердження
	незалежний пункт затвердження і архівізації даних

Дана оцінка дозволила розвинути Дж. Мартіном основи теорії проектування корпоративних комп'ютерних мереж і методологію побудови різноманітних проєкцій їх моделей.

В той же час дана оцінка одиниці руху даних не дозволяє врахувати ефективність використання ресурсів в пунктах формування, обробки та реєстрації даних, що не дозволяє реалізувати оптимізаційне проектування комп'ютерних мереж та розрахунок характеристик їх надійності, живучості, ймовірності перевантажень та відмов.

Дана оцінка руху даних практично не може бути використана для проектування та розрахунку системних характеристик мереж з глибоким розпаралеленням інформаційних потоків.

На основі коефіцієнта руху даних (7.1) можна визначити коефіцієнт ефективності руху даних, який враховує ресурси руху даних в конкретному вузлі матричної моделі

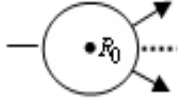
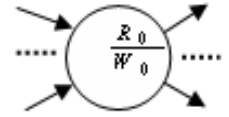
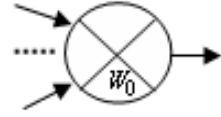
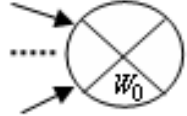
$$K_{ed} = \frac{R_i \cdot W_0}{R_0 \cdot W_i}, \quad (7.2)$$

де R_i, R_0, W_i, W_0 – відповідно фактичне число запитів, максимально можливе число запитів, фактичне число записів або оновлень, максимально можливе число записів або оновлень у вузлі матричної моделі.

У зв'язку з введенням описаної оцінки в характеристики двомірної матричної моделі необхідно ввести наступні позначення (табл. 7.2).

Таблиця 7.2

Позначення характеристик двомірної ММ

Тип вузла матричної моделі	Символ	Умова несуперечливості
Джерело даних		$R_0 > \sum_{i=1}^n R_i$
Пункт обробки даних		$R_0 > \sum_{i=1}^n R_i \quad W_0 > \sum_{j=1}^m R_j$
Залежний приймач даних		$W_0 > \sum_{j=1}^m R_j$
Незалежний приймач даних		$W_0 > \sum_{j=1}^m R_j$

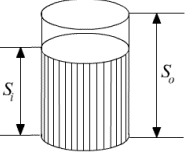
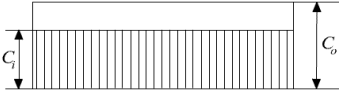

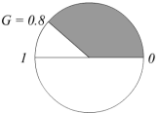
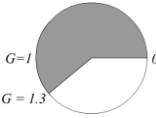
Запропоноване розширення атрибутів ММРД дозволяє розробити інженерну методику та інформаційну технологію проектування РКС на основі реальних топологій промислових підприємств.

7.2. Тримірні матричні моделі.

Інформаційна технологія побудови та теоретичне обґрунтування параметрів тримірної матричної моделі руху даних (ТММРД) базується на наступних принципах формалізації (табл.7.3).

Символіка тримірних матричних моделей руху даних, яка враховує коефіцієнт використання ресурсів елементів ММ, показана в табл.7.3.

Введена оцінка коефіцієнта ефективності руху даних (7.2) та символіка матричних моделей, яка представлена в табл.7.3, дозволяє розрахувати характеристики швидкості створення повідомлень у вузлах матричної моделі і шляхом їх представлення в тримірному просторі перейти до побудови тримірних матричних моделей.

Символ	Пояснення
	S_0 – максимальне число записів S_i – реальне число записів
	C_0 – швидкість створення та передавання даних C_i – проектна швидкість створення та передавання даних
	$G=1$ – завантаженість 100%
	$G=0.8$ завантаженість 80%
	$G=1.3$ 0.3 перевантаження по ресурсах читань (утворення черги)

На рисунку 7.2 показаний приклад тримірної проєкції двомірної матричної моделі.

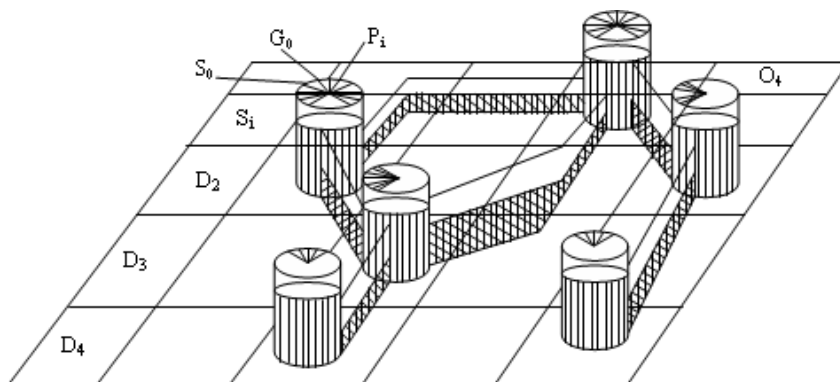


Рис.7.2. Тримірна проєкція двомірної матричної моделі руху даних.

ТММРД є ефективним способом представлення руху даних в КС, враховує ресурси та ступінь використання ресурсів КМ. Однак ТММРД не в повній мірі

відображає архітектуру та параметри руху даних в КМ, оскільки представлена досить громіздко в тримірному просторі і не дозволяє чітко представити часові атрибути матричної моделі.

7.3. Модифіковані двомірні матричні моделі.

Досвід розробки та використання ТММРД дозволяє представити її формальні параметри в двомірному варіанті ММ шляхом додаткового інформаційного насичення елемента ММ. На рис.7.3 показаний приклад інформаційного насичення елемента ДММ параметрами ТММРД.

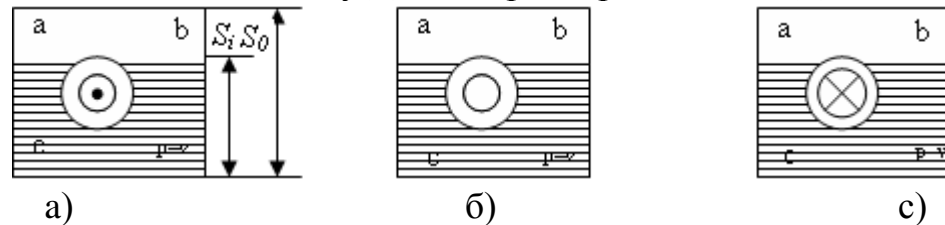


Рис.7.3. Інформаційне насичення елемента ДММ параметрами ТММРД.

На основі введених символів формуємо модифіковану двомірну ММ, яка враховує формальні параметри ТММРД (рис.7.4).

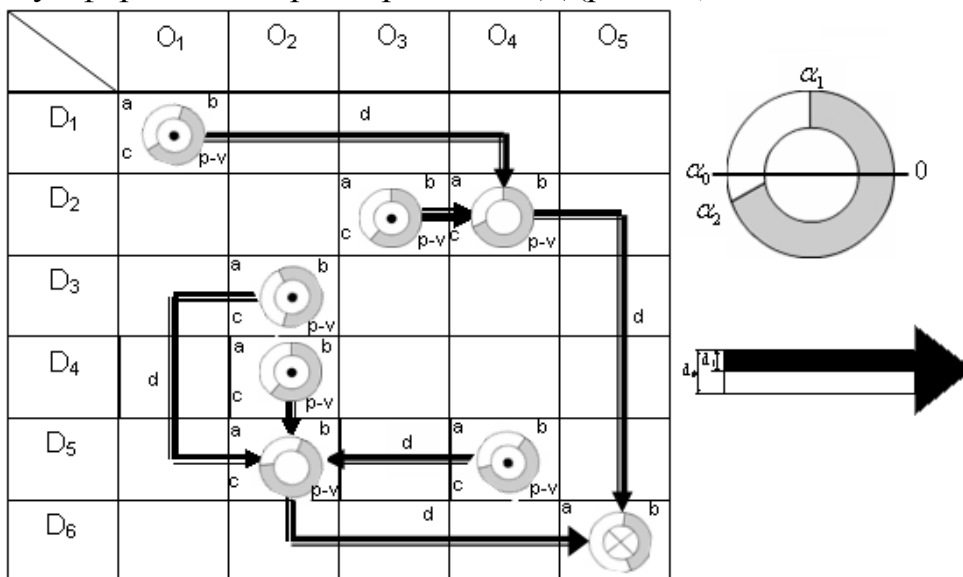


Рис.7.4. Модифікована двомірна ММ.

Запропонована система атрибутів та інформаційна технологія побудови модифікованих матричних моделей, приклад яких показаний на рис.7.4, де $\alpha_0 = 180^\circ$ – стовідсоткове використання ресурсів; $0 \leq \alpha_1 \leq 180^\circ$ – ступінь використання ресурсів зчитування даних; $180^\circ \leq \alpha_2 \leq 360^\circ$ – ступінь використання ресурсів записів; d_0, d_i – відповідно ресурси та ступінь використання швидкості передавання даних в каналі зв'язку.

Проведений аналіз інформаційних технологій побудови інформаційних моделей руху даних показує, що технологію проектування та аналіз КМ промислового підприємства в кожному конкретному випадку можна однозначно формалізувати на основі символіки ДММРД.

В той же час дана модель не відображає ресурсів КМ при виконанні операцій в кожному елементі ММ. Тому розроблена технологія побудови ТММРД, яка враховує наявність ресурсів в кожному елементі ММ, а також ступінь їх використання шляхом введення спеціальної символіки і трьохмірного зображення руху даних, компенсує недоліки ДММРД. Досвід проектування КМ та побудови ТММРД продемонстрував деяку громіздкість даного представлення руху даних у вигляді ТММРД. Розроблена технологія представлення ресурсів та їх ступені використання у вузлах ММ, шляхом модифікації ДММ привела до спрощення моделі представлення формалізованого руху даних без втрати інформативності.

Очевидно, що в окремих випадках можуть використовуватись всі типи описаних інформаційних моделей руху даних, на основі яких однозначно у формалізованій формі можуть бути побудовані інші проекції моделей руху даних, які входять в їх базове сімейство.

7.4. Аналіз топології промислового об'єкта управління.

На рис.7.5 представлений приклад топологічної схеми збору та цифрової обробки даних в низовій комп'ютерній мережі обліку витрат енергоносіїв.

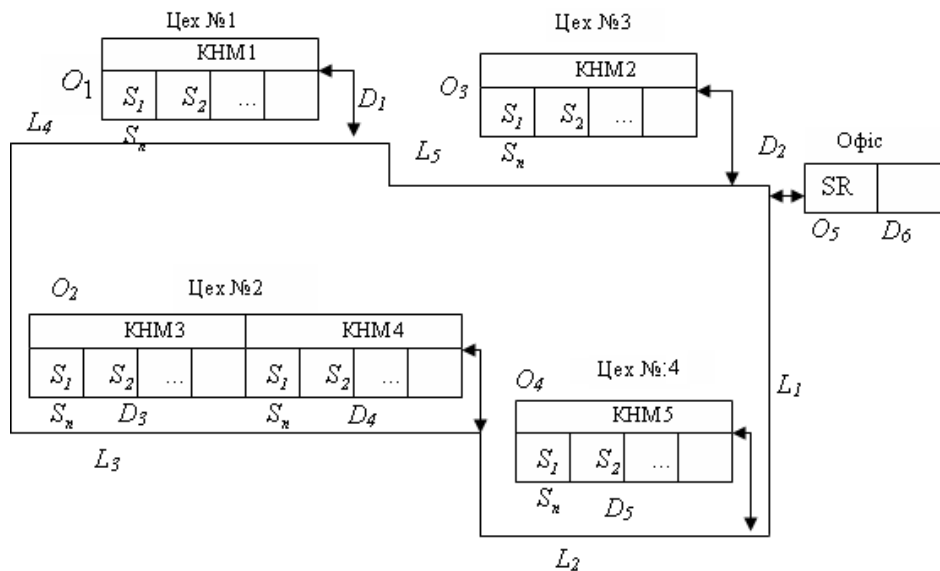


Рис.7.5. Топологія комп'ютерної мережі на підприємстві: SR–сервер, КНМ–контролер низової мережі, S₁– сенсори , L₁– довжини комунікаційних ліній мережі між окремими цехами.

З рис.7.5 видно, що модель руху даних повинна мати розмірність 5×6. Тобто цехи і офіс визначають об'єкти ММ O₁, O₂... O₅. У цехах підприємства та офісі

виникають відповідні класи даних, які формуються цеховими КНМ $D_1, D_2 \dots D_6$. Таким чином, представлена топологія КС може бути описана двомірною матричною моделлю (рис.7.6).

Слід зауважити, що на графі ММ точну прив'язку до об'єктів та даних мають джерела інформації та пункти затвердження і архівізації даних, а пункти обробки даних мають точну прив'язку до об'єктів.

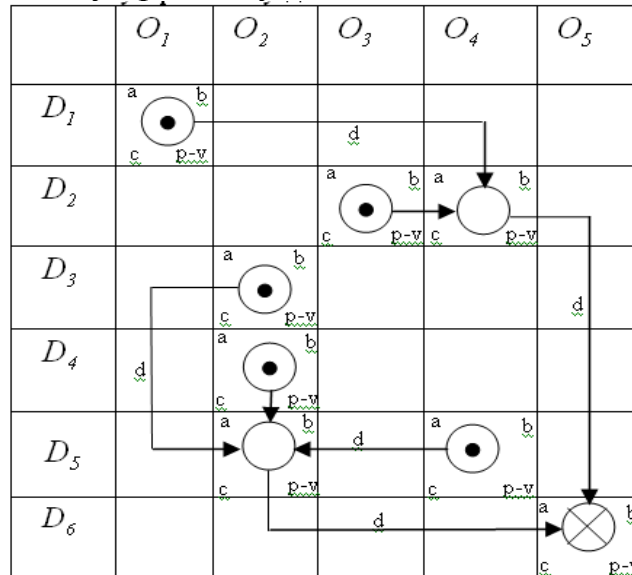


Рис.7.6. Двомірна ММ руху даних на підприємстві.

На основі формальних параметрів двомірної ММ будується наступна сукупність моделей руху даних:

- модель граф-розгалужене дерево;
- параметрична часова модель;
- структурно-часова модель;
- мережевий графік;
- суміщений часовий граф;
- блок-схема алгоритму обробки та контролю руху даних;
- граф-алгоритмічна модель.

Крім того, формалізуються епюри собівартості циклів руху даних (ЕРД): сигнальна прибутково-затратна ЕРД, диференціальна (Δ ЕРД), інтегральна ($\int \Delta \text{ЕРД}$), сумарна інтегральна ($\sum \int \Delta \text{ЕРД}$), глобальна GERД (G).

В той же час, незважаючи на досить детальне відображення руху даних в КМ у вигляді проекцій двомірних ММ на основі сімейства перелічених моделей, двомірна ММ не відображає наявні обчислювальні ресурси та степінь їх використання, що охоплюється тримірними ММ.

8. ПОХІДНІ МОДЕЛІ РУХУ ДАНИХ СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

8.1. Розробка похідних моделей на основі матричних моделей руху даних.

В основі побудови матричної моделі покладений двовимірний граф, приклад якого поданий на рис.8.1.

Матрична модель утворює мережу направлених зв'язків між джерелами інформації (ДІ), місцями обробки і приймачами даних, відображає маршрути документопотоків, що передаються по каналах зв'язку згідно топології інформаційної системи реальних промислових об'єктів, системи передачі даних або локальної комп'ютерної мережі.

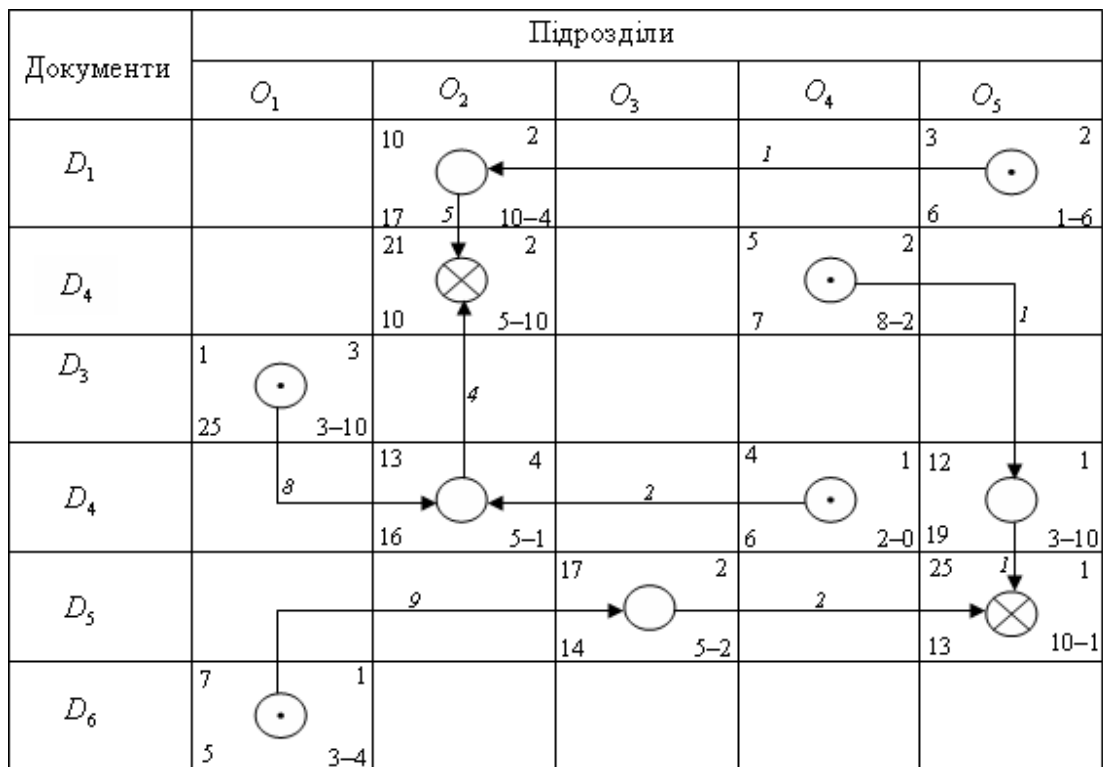


Рис.8.1. Матрична модель руху даних РКС.

Формалізацію матричної моделі проводять наступним чином:

$$D_i O_j \cdot X \cdot a, b, c, p - v \quad i \in \overline{1, n}, \quad j \in \overline{1, m}, \quad D_i O_j : D_j O_i,$$

$D_3O_1 \cdot 1 \cdot 1 \cdot 3 \cdot 25 \cdot 3 - 10;$
 $D_6O_2 \cdot 1 \cdot 7 \cdot 1 \cdot 5 \cdot 3 - 4;$
 $D_1O_2 \cdot 2 \cdot 10 \cdot 2 \cdot 17 \cdot 10 - 4;$
 $D_2O_2 \cdot 3 \cdot 21 \cdot 2 \cdot 10 \cdot 5 - 10;$
 $D_4O_2 \cdot 2 \cdot 13 \cdot 4 \cdot 16 \cdot 5 - 1;$
 $D_5O_3 \cdot 2 \cdot 17 \cdot 2 \cdot 14 \cdot 5 - 2;$
 $D_2O_4 \cdot 1 \cdot 5 \cdot 2 \cdot 7 \cdot 8 - 2;$
 $D_4O_4 \cdot 1 \cdot 4 \cdot 1 \cdot 6 \cdot 2 - 0;$
 $D_1O_5 \cdot 1 \cdot 3 \cdot 2 \cdot 6 \cdot 1 - 6;$
 $D_4O_5 \cdot 2 \cdot 12 \cdot 1 \cdot 19 \cdot 3 - 10;$
 $D_5O_5 \cdot 3 \cdot 25 \cdot 1 \cdot 13 \cdot 10 - 1;$

$D_3O_1 : D_4O_2$
 $D_6O_1 : D_5O_3$
 $D_1O_2 : D_2O_2$
 $D_4O_2 : D_2O_2$,
 $D_5O_3 : D_5O_5$
 $D_2O_4 : D_4O_5$
 $D_4O_4 : D_4O_2$
 $D_1O_5 : D_1O_2$
 $D_4O_5 : D_5O_5$

де $X=1, 2, 3$ відповідно ідентифікатор джерела, пункту обробки та приймача інформації.

ММ є базою для побудови всієї сукупності інформаційних моделей ОУ та МРД КС складних ДІ.

8.1.1. Граф - розгалужене дерево.

Побудова моделі типу граф – розгалужене дерево реалізує наявну мережеву структуру ММ у відповідний набір ієрархічних моделей з приймачем в якості кореня. При цьому досягається можливість просторового подання необхідних структур даної моделі, представлення всіх джерел та відображення шляхів руху даних, установлених на рівні конкретного приймача.

Формалізація моделі граф – розгалужене дерево:

$$\left. \begin{array}{l} D_3O_1 : D_4O_2 : D_2O_2 \\ D_4O_4 : D_4O_2 : D_2O_2 \\ D_1O_5 : D_1O_2 : D_2O_2 \end{array} \right\} \in D_2O_2 \cdot 3;$$

$$\left. \begin{array}{l} D_6O_1 : D_5O_3 : D_5O_5 \\ D_2O_4 : D_4O_5 : D_5O_5 \end{array} \right\} \in D_5O_5 \cdot 3.$$

Приклад такої моделі, побудованої на базі ММ (рис.8.1), поданий на рис.8.2.

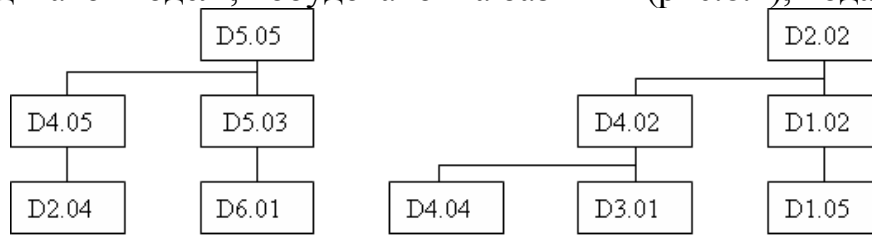


Рис.8.2. Модель “граф–розгалужене дерево”.

З рис.8.2. видно, що методологічно модель типу “граф– розгалужене дерево” будується розширенням графа вліво і вниз.

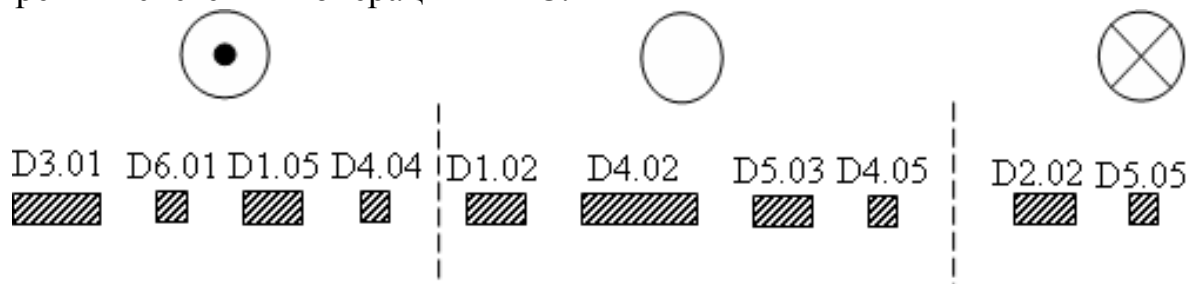
8.1.2. Часові інформаційні моделі.

Виділяють наступні часові інформаційні моделі:

- параметрична часова модель;
- структурно–часова модель;
- мережевий графік руху даних;
- суміщений часовий граф виконання системних функцій.

8.1.2.1. Параметрична часова модель.

Параметрична часова модель (рис.8.3) структурно показує необхідний час виконання системних операцій, згрупованих по ознаках джерел, пунктів обробки та приймачів інформації. Розглянута модель дозволяє оцінити і розрахувати необхідні часові, апаратні або людино–машинні ресурси для реалізації конкретних системних операцій в РКС.



Масштаб: – 1 доба.

Рис.8.3. Параметрична часова модель.

Формалізація параметричної часової моделі має наступний вигляд:

Таблиця ідентифікаторів активного вузла ММРД:

$$\begin{array}{ll}
 D_3O_1 \cdot 1 \cdot b_{31}; & D_5O_3 \cdot 2 \cdot b_{53}; \\
 D_6O_1 \cdot 1 \cdot b_{61}; & D_2O_4 \cdot 1 \cdot b_{24}; \\
 D_1O_2 \cdot 2 \cdot b_{21}; & D_4O_4 \cdot 1 \cdot b_{44}; \\
 D_2O_2 \cdot 3 \cdot b_{22}; & D_1O_5 \cdot 1 \cdot b_{15}; \\
 D_4O_2 \cdot 2 \cdot b_{42}; & D_4O_5 \cdot 2 \cdot b_{45}; \\
 & D_5O_5 \cdot 3 \cdot b_{55}.
 \end{array}$$

Впорядкована таблиця активних атрибутів ММРД з ознакою тривалості виконання операції b_{ij} :

$$\begin{array}{lll}
 D_3O_1 \cdot 1 \cdot b_{31}; & & \\
 D_6O_1 \cdot 1 \cdot b_{61}; & & \\
 D_2O_4 \cdot 1 \cdot b_{24}; & & \\
 D_4O_4 \cdot 1 \cdot b_{44}; & & \\
 D_1O_5 \cdot 1 \cdot b_{15} & & \\
 D_1O_2 \cdot 2 \cdot b_{12}; & & \\
 D_4O_2 \cdot 2 \cdot b_{42}; & & \\
 D_5O_3 \cdot 2 \cdot b_{53}; & & \\
 D_4O_5 \cdot 2 \cdot b_{45}; & & \\
 D_2O_2 \cdot 3 \cdot b_{22}; & & \\
 D_5O_5 \cdot 3 \cdot b_{55}. & &
 \end{array}$$

8.1.2.2. Структурно–часова модель.

Структурно–часова модель (рис.8.4) також визначає групування системних операцій по джерелах, пунктах обробки та приймачах, з прив'язкою по горизонталі до початку виконання системних процедур і зберіганням структури руху даних, а по вертикалі – з прив'язкою до конкретних джерел і приймачів даних. Дана модель відображає часову послідовність (причинність) системних процедур і дозволяє на стадії проектування або модернізації розкрити часові неузгодження руху даних і узгодити ресурси мережі з структурою руху даних.

Формалізація параметрів структурно–часової моделі має вигляд:

$$D_1O_5 \cdot 1 : D_2O_2 \cdot 2 \wedge (D_3O_1 \cdot 1 \wedge D_4O_4 \cdot 1 : D_4O_2 \cdot 2) : D_2O_2 \cdot 3;$$

$$(D_2O_4 \cdot 1 : D_4O_5 \cdot 2 \wedge D_6O_1 \cdot 1 : D_5O_3 \cdot 2) : D_5O_5 \cdot 3;$$

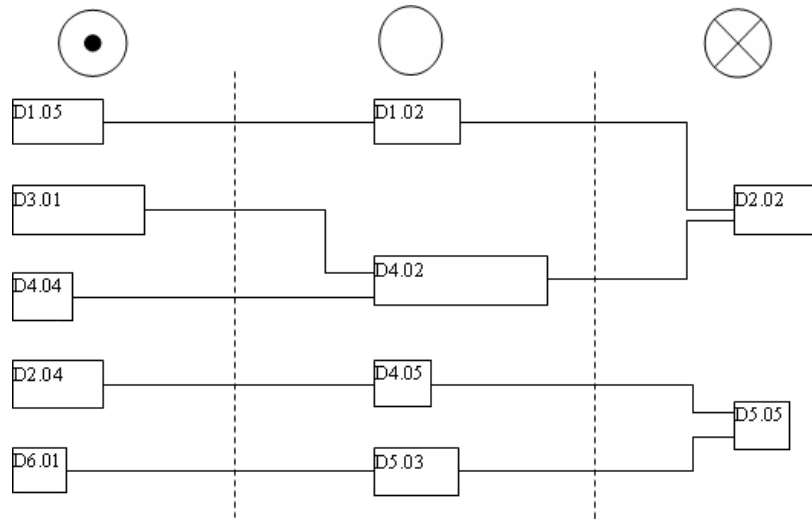


Рис.8.4. Структурно–часова модель

8.1.2.3. Модель “мережвий графік”.

Модель типу ”мережвий графік” (рис.8.5.) будується по типовій методиці побудови мережвих графіків і мереж. В даній моделі з часових параметрів враховується тільки час початку виконання системної операції (a).

Формалізація моделі мережвий графік передбачає формування:

Матриця ідентифікаторів активних вузлів початку виконання операції (a_{ij}):

$$\begin{array}{ll} D_3O_1 \cdot 1 \cdot a_{31}; & D_5O_3 \cdot 2 \cdot a_{53} \\ D_6O_1 \cdot 1 \cdot a_{61}; & D_2O_4 \cdot 1 \cdot a_{24} \\ D_1O_2 \cdot 2 \cdot a_{21}; & D_4O_4 \cdot 1 \cdot a_{44} \cdot \\ D_2O_2 \cdot 3 \cdot a_{22}; & D_1O_5 \cdot 1 \cdot a_{15} \\ D_4O_2 \cdot 2 \cdot a_{42}; & D_4O_5 \cdot 2 \cdot a_{45} \\ & D_5O_5 \cdot 3 \cdot a_{55} \end{array}$$

Впорядкована таблиця активних атрибутів з ознакою a_{ij} :

$D_3O_1 \cdot 1 \cdot a_{31};$	$D_1O_2 \cdot 2 \cdot a_{12}$
$D_6O_1 \cdot 1 \cdot a_{61};$	$D_4O_2 \cdot 2 \cdot a_{42}$
$D_2O_4 \cdot 1 \cdot a_{24}$	$D_5O_3 \cdot 2 \cdot a_{53}$
$D_4O_4 \cdot 1 \cdot a_{44};$	$D_4O_5 \cdot 2 \cdot a_{45}$
$D_1O_5 \cdot 1 \cdot a_{15};$	$D_2O_2 \cdot 3 \cdot a_{22}$
	$D_5O_5 \cdot 3 \cdot a_{55}$

Таблиця направлених зв'язків від джерел до приймачів:

$$D_3O_1 \cdot 1 \cdot a_{31} : D_4O_2 \cdot 2 \cdot a_{42} : D_2O_2 \cdot 3 \cdot a_{22}$$

$$D_6O_1 \cdot 1 \cdot a_{61} : D_5O_3 \cdot 2 \cdot a_{53} : D_5O_5 \cdot 3 \cdot a_{55}$$

$$D_4O_4 \cdot 1 \cdot a_{44} : D_4O_2 \cdot 2 \cdot a_{42} : D_2O_2 \cdot 3 \cdot a_{22}$$

$$D_1O_5 \cdot 1 \cdot a_{15} : D_1O_2 \cdot 2 \cdot a_{12} : D_2O_2 \cdot 3 \cdot a_{22}$$

$$D_2O_4 \cdot 1 \cdot a_{24} : D_4O_5 \cdot 2 \cdot a_{45} : D_5O_5 \cdot 3 \cdot a_{55}$$

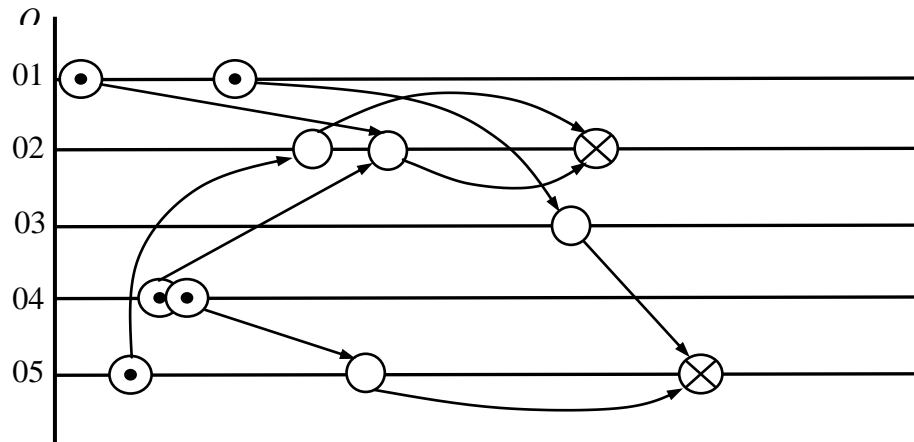


Рис.8.5. Мережевий графік.

8.1.2.4. Модель "суміщений часовий граф".

Модель типу "суміщений часовий граф" (рис.8.6) показує асоціацію системних функцій з заданими часовими параметрами без структури руху даних. Розглянута модель дозволяє розрахувати поточні навантаження на обчислювальні ресурси людино-машинних засобів обробки даних в кожному елементі ММ або розподілення пікових навантажень всією мережею обчислювальної системи в цілому. Призначена вона для центрального сервера КС, який контролює регламентність виконання функцій підрозділами КС і координує їх роботу в часі.

Формалізація моделі "суміщений часовий граф":

$$f_i - D_i O_j \cdot a_{ij} \cdot a_{ij} + b_{ij};$$

$$\begin{aligned}
 f_1 &= D_3 O_1 \cdot 1 \cdot 1 + 3; & f_6 &= D_1 O_2 \cdot 10 \cdot 10 + 2; \\
 f_2 &= D_1 O_5 \cdot 3 \cdot 3 = 2; & f_7 &= D_4 O_5 \cdot 12 \cdot 12 + 1; \\
 f_3 &= D_4 O_4 \cdot 4 \cdot 4 + 1; & f_8 &= D_4 O_2 \cdot 13 \cdot 13 + 4; \\
 f_4 &= D_2 O_4 \cdot 5 \cdot 5 + 2; & f_9 &= D_5 O_3 \cdot 17 \cdot 17 + 2; \\
 f_5 &= D_6 O_1 \cdot 7 \cdot 7 + 1; & f_{10} &= D_2 O_2 \cdot 21 \cdot 21 + 2; \\
 & & f_{11} &= D_5 O_5 \cdot 25 \cdot 25 + 1.
 \end{aligned}$$

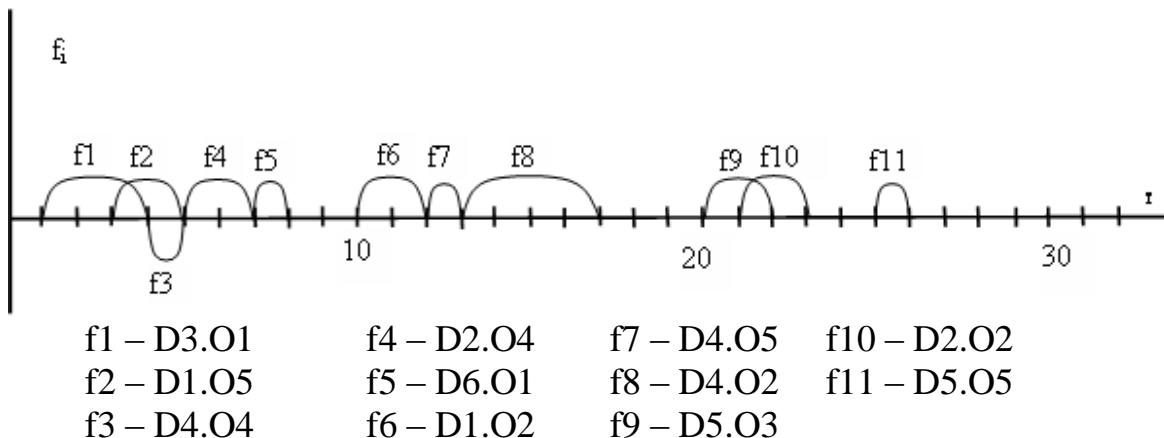


Рис.8.6. Модель "суміщений часовий граф".

8.1.3. Модель "блок–схема алгоритму опрацювання даних".

Модель "блок–схема алгоритму опрацювання даних" (рис.8.8) є діагностичною моделлю КС, яка будується на основі суміщеної часової моделі у відповідності з використанням типових позначень блок–схем алгоритмів і програм. Дана модель використовується центральним сервером РКС для діагностування виконання процесів руху даних в системі згідно регламентного алгоритму її функціонування на основі моделі „суміщений часовий граф”.

Суть побудови даної моделі методично виконується формалізацією процедур побудови оптимального несуперечного змістовного графа розгалуженого алгоритму у наступному порядку:

- формалізація умови задачі;
- побудова суміщеного часового графа;
- побудова логічного розгалуженого графа;
- покриття логічного графа блок–схемою;
- нумерація операторів блок–схеми.

Формалізація моделі „блок–схема алгоритму”:

$$f_i = \begin{cases} f_1, T_1 \leq T \leq T_4; \\ f_2, T_3 \leq T \leq T_5; \\ f_3, T_4 \leq T \leq T_5; \\ f_4, T_5 \leq T \leq T_7; \\ f_5, T_7 \leq T \leq T_8; \\ f_6, T_{10} \leq T \leq T_{12}; \\ f_7, T_{12} \leq T \leq T_{13}; \\ f_8, T_{13} \leq T \leq T_{17}; \\ f_9, T_{20} \leq T \leq T_{22}; \\ f_{10}, T_{21} \leq T \leq T_{23}; \\ f_{11}, T_{25} \leq T \leq T_{26}. \end{cases}$$

Формалізація умови задачі полягає у відповідності до нумерації системних функцій $f_i(r)$, які в реальній процедурі описуються системою аналітичних виразів, які виконуються при заданих часових організаціях.

Наприклад, для умови задачі $f_i(r)$ – багатоканальне аналого–цифрове перетворення технологічних параметрів; $f_2(t_2)$ – ковзне усереднення формуючих відліків процесів; $f_3(t_3)$ – обчислення матриці коефіцієнтів кореляції; $f_4(t_4)$ – індикація параметрів.

Формалізація умови задачі побудови моделі „блок-схеми алгоритму”:

$$Y(t) = \begin{cases} f_1(t) & d \leq t < b; \\ f_2(t) & c \leq t \leq d; \\ f_3(t) & e < t. \end{cases} ,$$

де a, b, c, d, e – часові обмеження.

Нехай $a=1, b=2, c=4, d=6, e=5$. Тоді суміщений часовий граф (СЧГ) має вигляд, приведений на рис.8.7, 8.8, де стрілки вказують, що відповідні системні процедури виконуються до настання часу $t = b$ або пізніше часу $t = e$.

Змінюючи значення часових обмежень a, b, c, d, e і здійснюючи розпаралелювання операцій виконання системних процедур $f_i(t)$, одержимо відповідно різні реалізації суміщеного графа (рис. 8.8).

Для визначення формалізованої методики побудови розгалуженого логічного графа (РЛГ) сформулюємо ряд тверджень:

1. Системними атрибутами логічного графа є 5 вершин: початок, ввід–вивід, оператор системної функції, умова і кінець;
2. Основним атрибутом розгалуженого логічного графа є умова, при чому:
 - 2.1) якщо умова виконується, ЛГ розширюється вправо, в протилежному випадку – вниз;
 - 2.2) якщо умова не виконується, то вимагається її уточнення, граф розширюється зліва вниз;
3. Вид суміщеного часового графа одночасно визначає структуру логічного графа, при чому:

3.1) якщо системні функції на СЧГ не перетинаються і не накладаються, то ЛГ розширюється вправо і вниз, а для виводу використовується одна загальна вершина;

3.2) в протилежному випадку ЛГ розширюється тільки вниз, а для виводу використовується автономна вершина після кожного оператора системної функції.

Приклади побудови розгалужених ЛГ та блок-схем алгоритмів для суміщених часових моделей показані на рис.8.7(a) – 8.7(г).

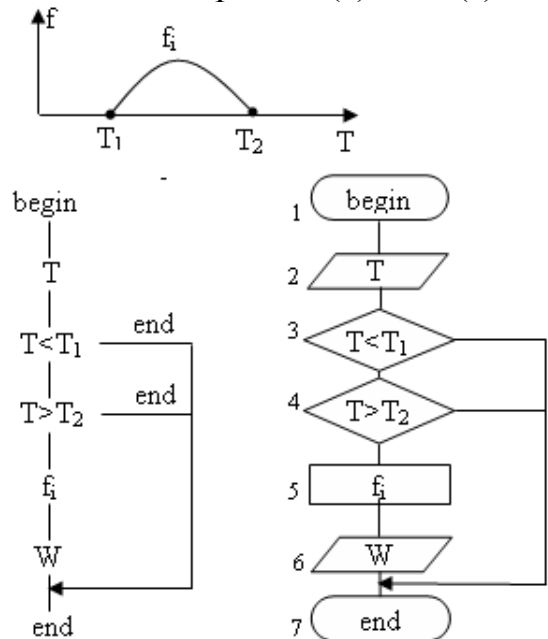


Рис. 8.7(а). Граф та блок-схема алгоритму.

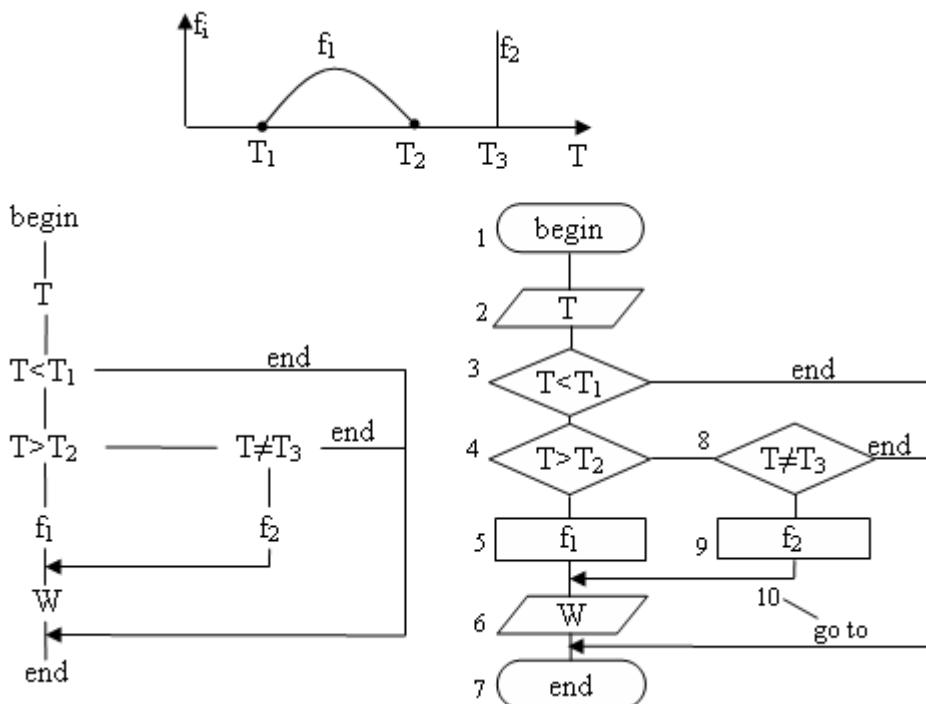


Рис. 8.7(б). Граф та блок-схема алгоритму.

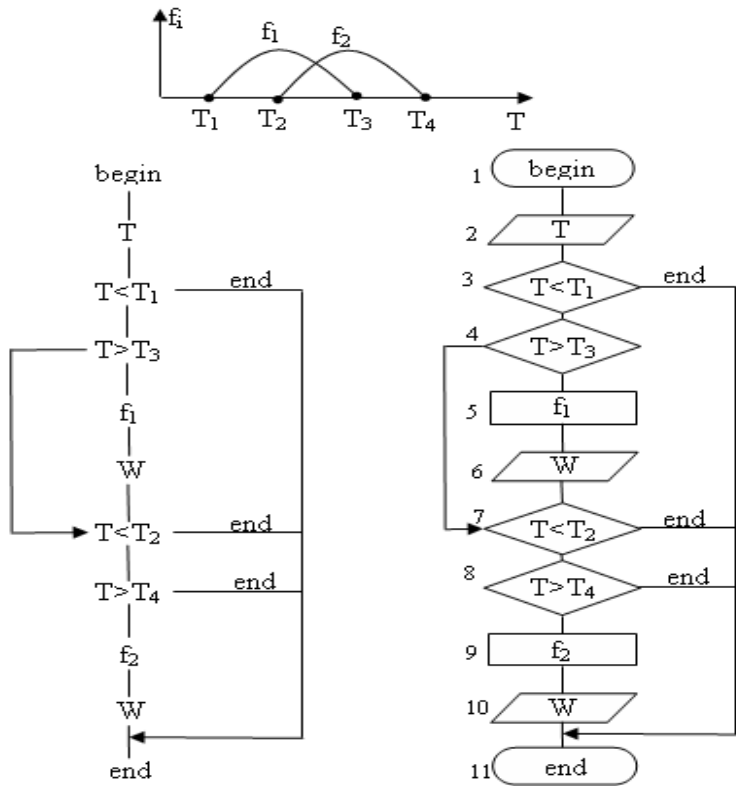


Рис. 8.7(в). Граф та блок-схема алгоритму.

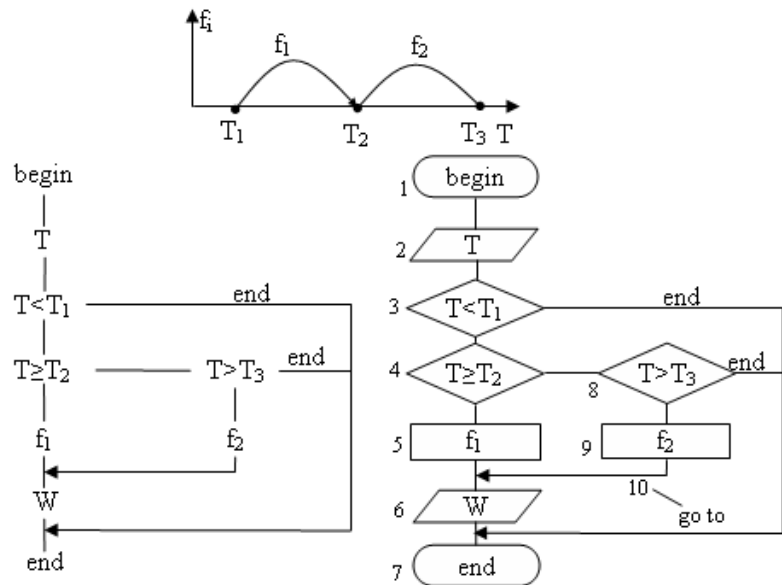


Рис. 8.7(г). Граф та блок-схема алгоритму.

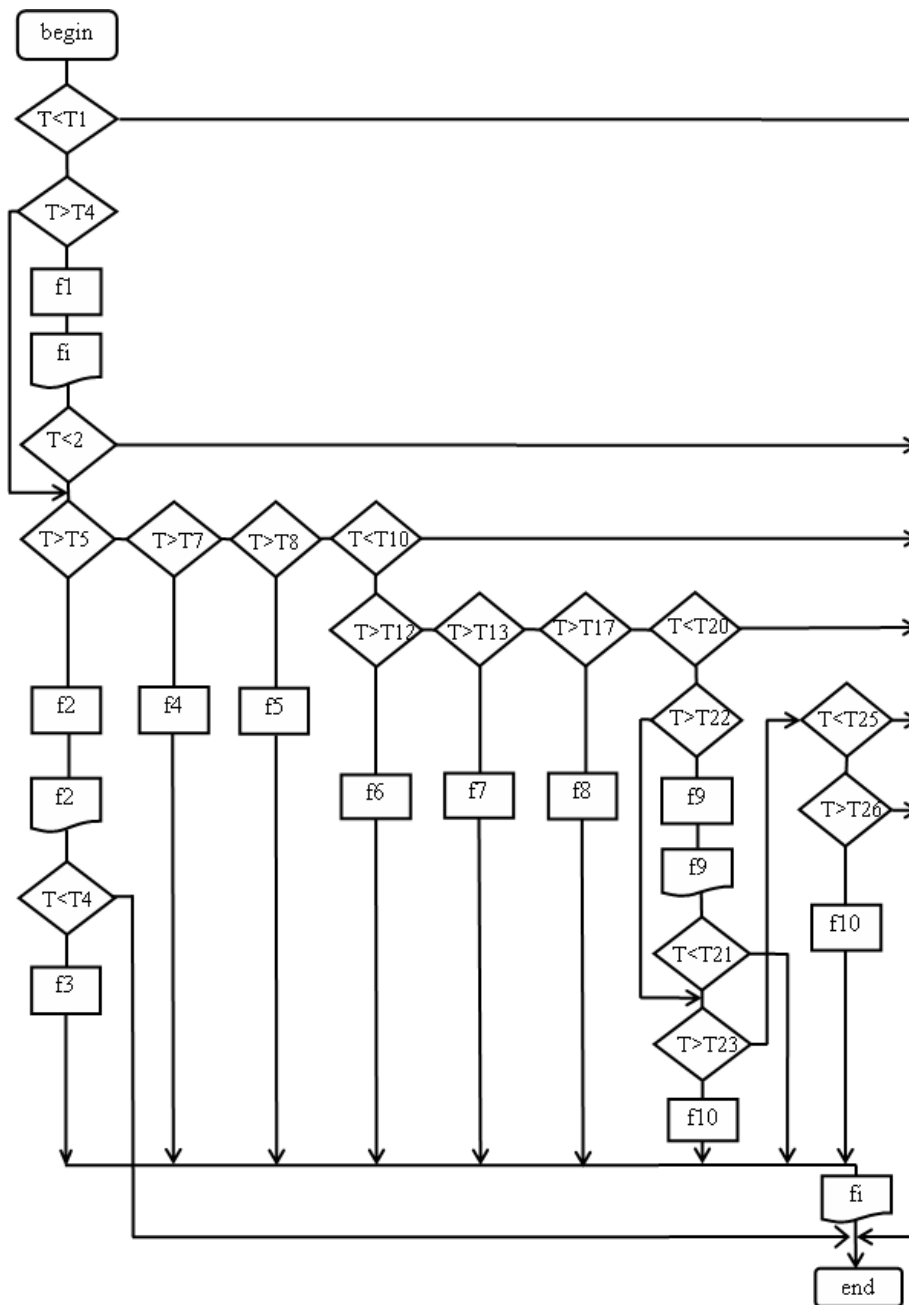


Рис.8.8. Блок-схема алгоритму оброблення даних.

8.1.4. Граф-алгоритмічна модель опрацювання даних.

Побудова граф-алгоритмічної моделі (ГАМ) виконується у вигляді змістовного символічного графу, який будується згідно архітектури зв'язків ММРД з символікою типів операцій формування, обробки, приймання та зберігання даних. Побудова даної моделі виконується зверху – вниз для кожного джерела та приймача, який розміщений на відповідних об'єктах.

Приклад побудови граф-алгоритмічної моделі на основі матричної моделі (рис.8.1) та згідно символіки (табл.8.1) показано на рис.8.9

Таблиця 8.1

№ п/п	Позначення	Зміст позначення
1		Неавтономна пам'ять
2		Автономна пам'ять
3		Тверда копія документа
4		Звуковий ввід/вивід
5		Мишка
6		Магнітно-оптичний диск
7		Магнітна дискета
8		Оптичний диск
9		Магнітний диск
10		Архів
11		Магнітна карта
12		Модем
13		Дисплей
14		Процес
15		Рішення
16		Модифікація
17		Конкретизований процес
18		Ручна операція
19		Допоміжна операція
20		Ручний ввід

21		Оптичний канал
22		Ввід-вивід
23		Пуск-зупинка
24		Ручний документ
25		Оператор
26		Джерело-приймач
27		Злиття
28		Виділення
29		Групування
30		Сортування
31		Сторінковий з'єднувач
32		Міжсторінковий з'єднувач
33		Безпосередня передача
34		Інформаційний потік
35		Дублювання передачі
36		Канал зв'язку
37		Матеріальний потік
38		Розшифрування
39		Кодування
40		Копіювання
41		Автономна обробка

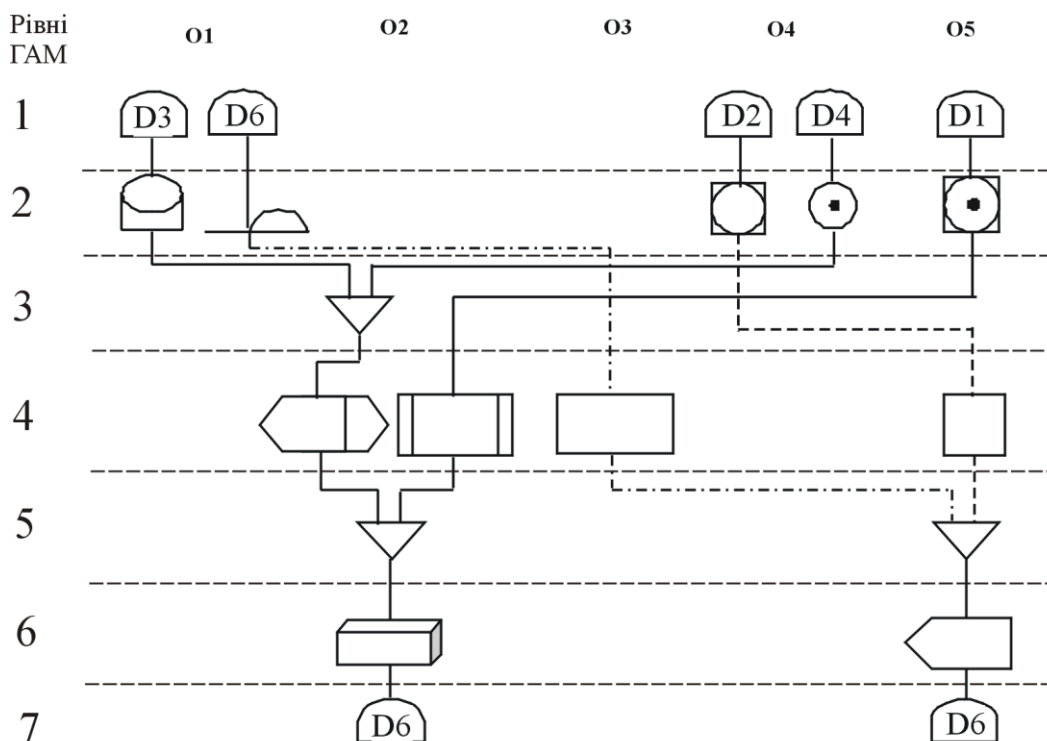


Рис.8.9. Граф-алгоритмічна модель

З рис.8.9 видно, що в першому підрозділі розміщені 2 джерела інформації, які формують дані типу D3 і D6. Відповідно в підрозділах 04 та 05 розміщені джерела інформації, що формують дані типу D2, D4 та D1, що представлені на першому рівні ГАМ. На другому рівні розміщена група символів при поширенні граф-алгоритмічної моделі вниз, яка демонструє фізичні носії даних, згідно номерів в нижній лівій частині елемента матричної моделі, які відповідають таблиці стандартизованих символів МРД. На 3 та 5 рівнях ГАМ показані символи об'єднання інформаційних потоків в комп'ютерній мережі з демонстрацією позначень різних типів каналів зв'язку. На 4 та 6 рівнях ГАМ – символи типів операцій, які задаються відповідними номерами в ММ в нижній лівій частині елементів, які відповідають пунктам обробки та затвердження операції. На 7 рівні показані символи приймачів інформації, які розміщені в підрозділах 02 та 05. Показаний рівень ГАМ може бути представлений відповідною архітектурою більш низького рівня ГАМ (наприклад, підприємство – цех, цех – технологічна установка, технологічна установка – оператор).

Контрольні запитання і завдання.

1. Які похідні моделі будуються формалізованим способом на основі матричних моделей руху даних?
2. Поясніть технологію побудови моделі "граф-розгалужене дерево".
3. Назвіть часові моделі руху даних.

4. Що характеризує параметрична часова модель руху даних?
5. Поясніть технологію побудови структурно-часової моделі.
6. Яким атрибутами описується модель "мережевий графік"?
7. Дайте приклад побудови моделі "суміщений часовий граф".
8. Які атрибути використовуються для побудови моделі "блок-схема алгоритму опрацювання даних"?
9. Дайте приклад побудови логічного графа на основі моделі "суміщений часовий граф".
10. Побудуйте блок-схему алгоритму на основі логічного графа руху даних з використанням стандартних графічних символів.
11. Наведіть приклад побудови граф-алгоритмічної моделі на основі матричної моделі.

9. ЕКОНОМІЧНІ ЕПЮРИ ЦИКЛІВ РУХУ ДАНИХ СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

9.1. Методи побудови епюр руху даних (ЕРД).

В даний час стало очевидним, що розвиток теорії проектування комп'ютерних мереж, особливо таких, які характеризуються паралельними інформаційними потоками, потребує відповідного розвитку теорії економічних епюр собівартості для вузлів матричних моделей таких мереж, а також створення інтегральних епюр собівартості витрат і прибутків, при проектуванні комп'ютерних систем, представлених тримірними матричними моделями.

Для визначення поняття економічних епюр (ЕЕ) розглянемо елемент двовимірної матричної моделі руху даних в комп'ютерній мережі (рис.9.1.)

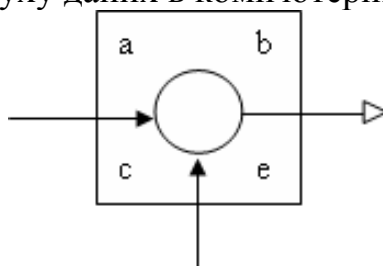


Рис.9.1. Елемент вузла матричної моделі руху даних:

a, b – відповідно час початку та тривалість виконання операції формування, обробки та затвердження даних;

● – символ джерела, ○ – пункту обробки, ⊗ – затвердження даних ;

c – тип операції у вузлі ММ;

e – собівартість виконання операції, формування, обробки та затвердження даних.

Під поняттям оцінки значення ЕЕ будемо розуміти відношення:

$$K_p = \frac{(P_{ij} - V_{ij})}{T_{ij}}, \quad (9.1)$$

де P_{ij} – сумарний дохід від впровадження функцій вузла ММ;

V_{ij} – затрати на постановку та запуск функцій вузла ММ;

T_{ij} – регламентні затрати часу на реалізацію функцій вузла ММ;

$i = 1, 2, \dots, n$ – число документів ММ;

$j = 1, 2, \dots, m$ – число підрозділів ММ;

Для ММ, яка представлена на рис.8.1 виконується побудова ЕРД наступного типу:

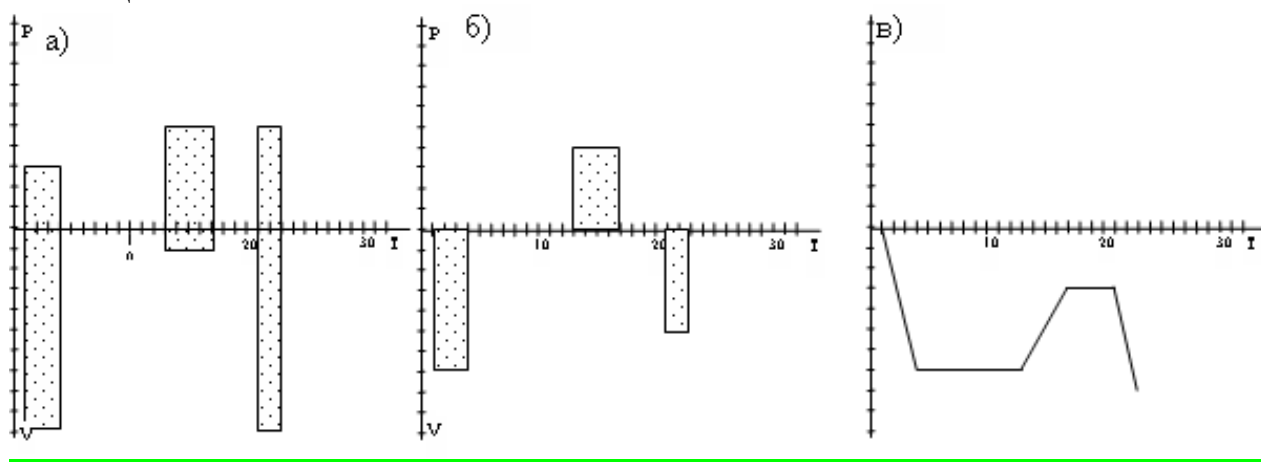
– цикли руху даних (рис.9.2) ґрунтуються на описі всіх трафіків руху даних від джерел до пунктів затвердження та зберігання даних;



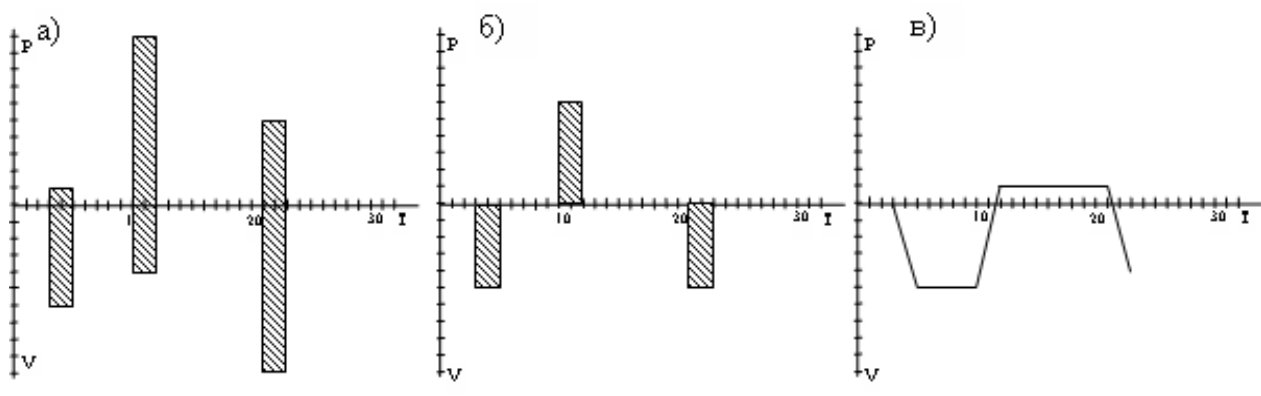
Рис. 9.2. Цикли руху даних.

- епюри собівартості циклу ЕРД (рис.9.3 (а)) будуються для кожного циклу руху даних на основі часових параметрів (a, b) та параметрів собівартості $(p-v)$ матричної моделі;
- диференціальна епюра ΔEPD руху даних (рис.9.13 (б)) представляє собою різницевий граф прибутків та витрат для кожного циклу руху даних;
- інтегральна епюра $(\int \Delta \dot{P} \dot{D} \dot{A})$ циклу руху даних (рис.9.13 (в)) будується на основі моделі ΔEPD шляхом інтегрування характеристик витрат та прибутків у кожному пункті руху даних матричної моделі (рис.9.12).

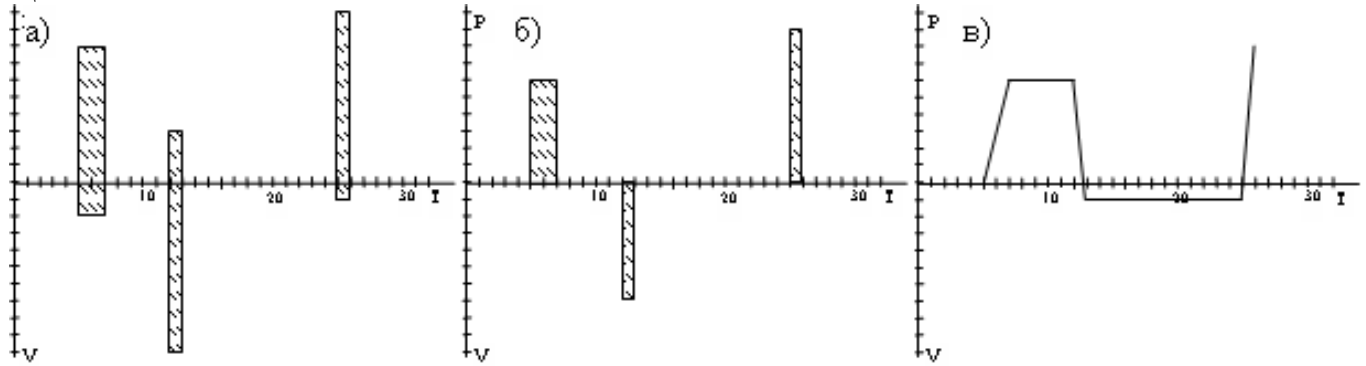
Цикл D3.O1–D4.O2–D2.O2



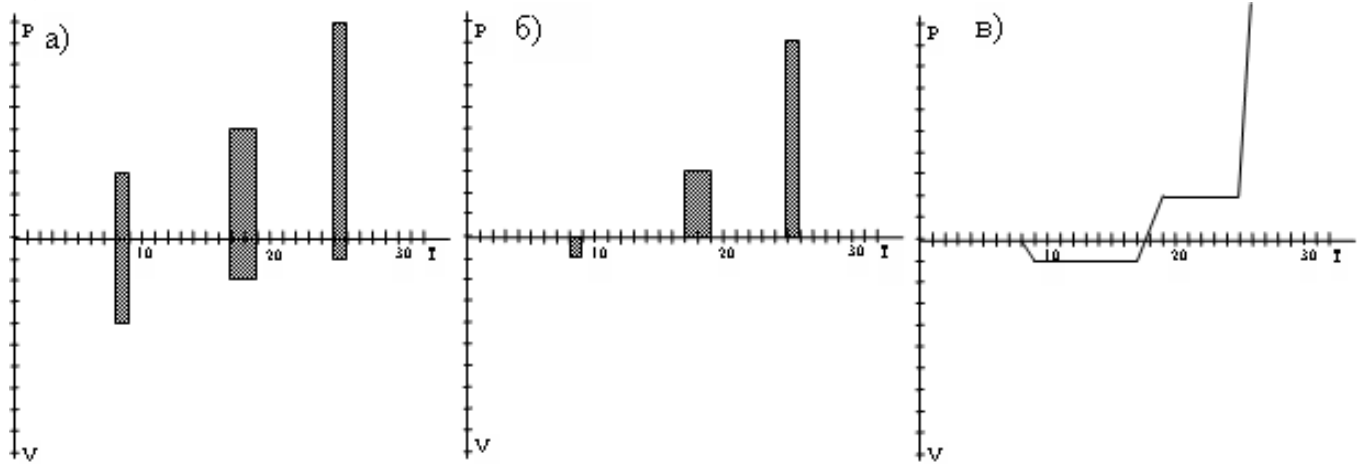
цикл D1.O5–D1.O2–D2.O2



Цикл D2.O4–D4.O5–D5.O5



Цикл D6.O1–D5.O3–D5.O5



Цикл D4.O4–D4.O2–D2.O2

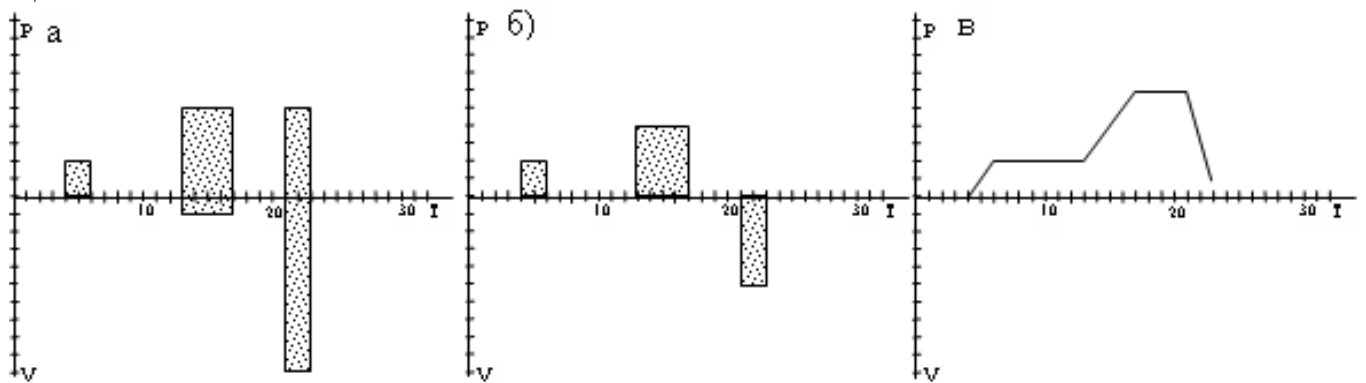


Рис.9.3. Епюри собівартості (а), диференціальні епюри (б), інтегральні епюри (в).

– сумарна інтегральна епюра ($\sum \int \Delta \dot{A} \dot{A}$) циклу руху даних будується шляхом сумування інтегральних епор (рис 9.4);

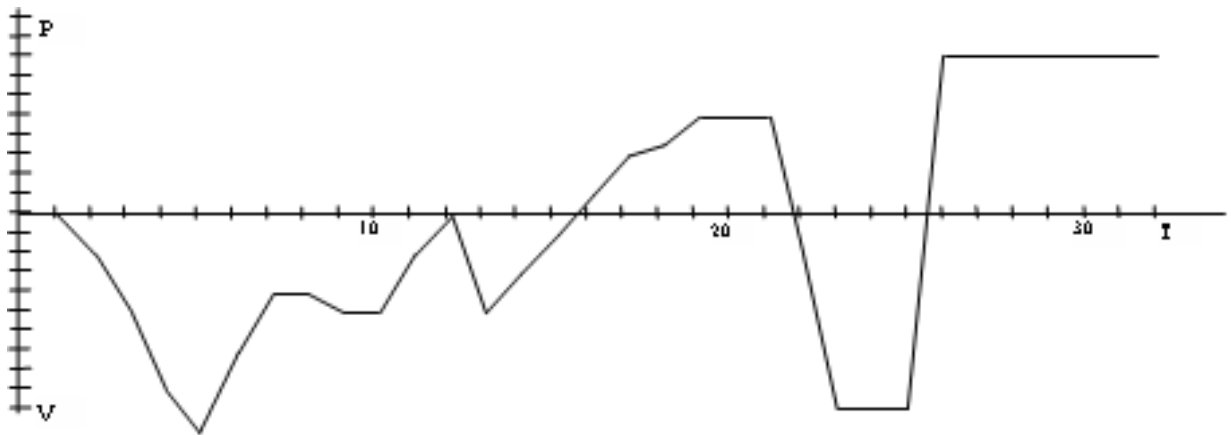


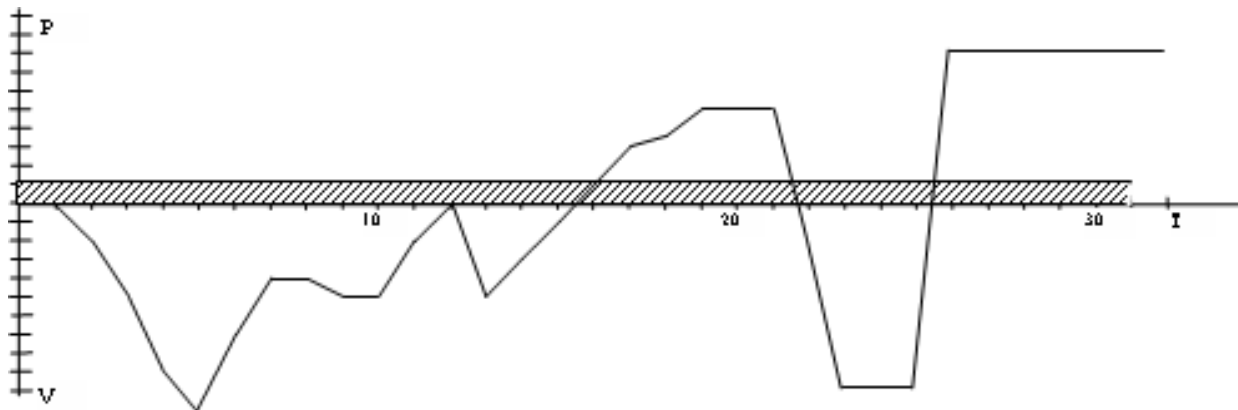
Рис. 9.4. Сумарна інтегральна епюра собівартості циклів руху даних.

На основі вищевказаних епюр будуюмо глобальну характеристику ефективності КСУ (рис.9.5).

Глобальна характеристика ефективності проектованої спеціалізованої комп'ютерної системи розраховується на основі обчислення вибіркового математичного сподівання характеристики сумарної інтегральної економічної епюри згідно формули:

$$G_{\hat{N}\hat{E}\hat{N}} = \frac{\sum_{i=1}^n \int_0^T \Delta \hat{A} \hat{D} \hat{A}_i(T) dT}{N},$$

де n – число часових інтервалів ковзної вибірки,
 N – загальне число часових інтервалів.



G_{kc}

Рис.9.5. Глобальна характеристика ефективності спеціалізованої комп'ютерної системи.

Викладені теоретичні основи, методи та інформаційні технології побудови епюр собівартості руху даних в КСУ є ефективним інструментом аналізу та діагностики діючих систем, а також засобом проектування складних РКС з високим рівнем розпаралелення системних операцій та основою реалізації стратегії вибору оптимізованих варіантів їх архітектури, трафіків інформаційних потоків, функціональних можливостей та ступеня використання ресурсів в активних вузлах ММ.

Контрольні запитання і завдання.

1. Поясніть технологію побудови епюр руху даних з послідовними та паралельними алгоритмами опрацювання даних.
2. Що таке цикл руху даних в КСУ?
3. Поясніть технологію побудови сигнальних епюр руху даних.
4. Як будуються диференціальні епюри руху даних на основі інтегральних?
5. Як будується сумарна інтегральна епюра собівартості циклів руху даних?
6. Як обчислюється і що відображає глобальна характеристика ефективності КСУ?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Автоматизовані системи обробки економічної інформації: Підручник / Г.В. Лавінський і ін. – К.: Вища школа, 1985.- 287 с.
2. Васильев В.В., Кузьмук В.В. Сети Петри, параллельные алгоритмы и моделирование мультипроцессорных систем. – К.: Наукова думка, 1990.
3. Возна Н.Я. Дослідження ефективності розподілених інформаційних систем на основі епюр собівартості циклів руху даних // Наукові вісті інституту менеджменту та економіки «Галицька академія». – Івано-Франківськ. – 2006. – №2(10). – С.74-78.
4. Возна Н.Я. Інформаційна технологія побудови граф-алгоритмічної моделі руху техніко-економічних даних // Збірник наукових праць інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України. – Випуск 39. – Київ. – 2006. – С. 44-51.
5. Возна Н.Я. Методологія та інформаційна технологія побудови моделей "блок-схема алгоритму руху даних" в розподілених комп'ютерних системах реального часу. Збірник наукових праць Буцацького інституту менеджменту і аудиту. – Бучач. – 2011 - №7.-С.336-339.
6. Возна Н.Я. Методологія та техніка формування техніко-економічних даних в автоматизованих системах управління // Вісник Хмельницького національного університету. – 2005. – №4. – Т.2, Ч.1. – С. 131-133.
7. [Корнеев](#) И.К., [Ксандопуло](#) Г.Н., [Машурцев](#) В.А. Информационные технологии.: – М.: [Перспект](#), 2007. – 854 с.
8. Локазюк В.М., Поморова О.В., Домінов А.О. Інтелектуальне діагностування мікропроцесорних пристроїв та систем: Навч. посібник для вузів. Хмельницький.- 2001.–286с.
9. Мартин Дж. Вычислительные сети и распределенная обработка данных.– М.:Финансы и статистика, 1985.–256с.
10. Мартин Дж. Планирование развития автоматизированных систем.- М.: Финансы и статистика, 1984.- 169 с.
11. [Михайлова](#) Е.Е., [Рыбак](#) К.С., [Тюкачев](#) Н.А. Программирование в Delphi для начинающих.- М.: [ВНУ](#), 2007. – 672 с.
12. [Могилев](#) А.В., [Листрова](#) Л.В. Информация и информационные процессы. Социальная информатика - М.: [ВНУ](#), 2006. – 240 с.
13. Николайчук Я.М. Низові обчислювальні мережі: Учбовий посібник.–К:УМК ВО, 1990.– 64с.
14. Николайчук Я.М., Возна Н.Я., Пітух І.Р. Проектування спеціалізованих комп'ютерних систем / Навчальний посібник / - Тернопіль: ТзОВ "Тернограф". 2010. – 392с., іл.

15. Николайчук Я.М., Пітух І.Р., Возна Н.Я. Теорія моделей руху даних розподілених комп'ютерних систем / Монографія - Тернопіль: ТзОВ "Тернограф", 2008 – 216 с.
16. Пестриков В., Маслобоев А. Delphi на примерах. – Л.: КБП, 2005. – 496 с.
17. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 264 с.
18. Пітух І., Николайчук Я., Возна Н. Моделювання руху даних та методологія проектування комп'ютерної мережі з паралельними інформаційними потоками // Вісник Технологічного університету Поділля. Технічні науки. – Хмельницький. – 2004. – Т.2, Ч.1, №2.– С. 33-35.
19. Пітух І., Николайчук Я., Возна Н. Принципи побудови комп'ютерних мереж з глибоким розпаралелюванням інформаційних потоків на основі матричних моделей руху даних // Вісник НУ „Львівська політехніка”. Радіоелектроніка та телекомунікації. – 2004.- № 508. - С. 263–268.
20. Пітух І.Р. Моделі комп'ютерних мереж на основі інтегральних економічних епюр // Збірник наукових праць, Інститут проблем моделювання в енергетиці НАН України. – Київ. – 2004.– № 27. – С.81–86.
21. Пітух І.Р. Системні характеристики формальних об'єктів моделей руху даних в комп'ютерних мережах // Тези доповідей III Міжнар. конф. „PHOTONICS–ODS 2005”. – Вінниця. – 2005. – С. 60–61.
22. Сегін А.І., Возна Н.Я. Методології побудови інформаційних моделей руху даних Вісник Технологічного університету Поділля. - 2002. - №3. - Т1. - С.128-135.
23. [Советов](#) Б.Я. Моделирование систем. Практикум: Учебное пособие .- М.-Высш. шк., 2005.- 295 с.
24. Стеклов В.К., БеркманЛ.Н. Проектування телекомунікаційних мереж.– К.: Техніка, 2002.–792с.
25. Столлингс В. Структурная организация и архитектура компьютерных систем: Пер. с англ.–М.: Издательский дом "Вильямс", 2002.–896с.
26. Столлингс В. Современные компьютерные сети .– СПб Питер, 2003.– 783 с.
27. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы. - СПб.: Питер, 2003. - 880с.
28. Томашевський В.М. Моделювання систем.– К.: Видавнича група ВНУ, 2005.– 352с.
29. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем : Учебник для вузов .– СПб.: Питер, 2006.– 668 с.