



УКРАЇНА

(19) **UA** (11) **77050** (13) **U**
(51) МПК (2013.01)
G06F 7/00
G06Q 99/00

ДЕРЖАВНА СЛУЖБА
ІНТЕЛЕКТУАЛЬНОЇ
ВЛАСНОСТІ
УКРАЇНИ

(12) ОПИС ДО ПАТЕНТУ НА КОРИСНУ МОДЕЛЬ

<p>(21) Номер заявки: u 2012 08774</p> <p>(22) Дата подання заявки: 16.07.2012</p> <p>(24) Дата, з якої є чинними права на корисну модель: 25.01.2013</p> <p>(46) Публікація відомостей про видачу патенту: 25.01.2013, Бюл.№ 2</p>	<p>(72) Винахідник(и): Ткачов Віталій Миколайович (UA), Саваневич Вадим Євгенович (UA), Брюховецький Олександр Борисович (UA), Анненков Олександр Борисович (UA)</p> <p>(73) Власник(и): ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ, пр. Леніна, 14, м. Харків, 61166 (UA)</p>
---	---

(54) СПОСІБ ПАРАЛЕЛЬНОЇ ОБРОБКИ ЦИФРОВИХ ДАНИХ ІЗ ЗАПОБІГАННЯМ ВИНИКНЕННЮ КОЛІЗІЙ

(57) Реферат:

Спосіб паралельної обробки цифрових даних із запобіганням виникненню колізій включає операції зчитування, обробку даних. При цьому здійснюють безпомилкову паралельну багатоетапну обробку порцій та серій порцій даних із формуванням повідомлень-запитів та стану у відповідності до обробника даних.

UA 77050 U

Корисна модель належить до області обчислювальної техніки і може бути використаний для цифрової обробки масивів даних в реальному масштабі часу.

Відомий спосіб обчислення та обробки даних (пат. Росії № 2142158С1, МПК⁷ G06F 15/00, G06F 15/16, публ. 27.11.1999) призначений для реалізації можливості обчислювальної станції
5 приєднатися до обчислювальної задачі в будь-який час і виконувати свою власну обчислювальну функцію зі своєю швидкістю та покладений в основу системи, яка описана в патенті.

Недоліком даного способу є низька оперативність, так як вірогідність звернення обчислювальної станції до задачі створює проблему колізії, рішень якої не вказано.

10 Другий відомий спосіб обчислення і обробки даних (пат. Росії № 2145438С1, МПК⁷ G06F 17/60, публ. 10.02.2000) призначений для комерційних і управлінських специфічних функцій обліку і звітності, що включає операції зчитування вихідних даних виділення даних і їх перетворення.

15 Недоліком даного способу є жорсткість масштабованості та відсутність можливостей інтеграції з іншими подібними системами.

Відомий також, вибраний як найближчий аналог, спосіб обробки цифрових даних (пат. України № 47741 А, МПК (2006) G06F 7/00, G06F 7/38, G06Q 99/00, публ. 15.07.2002, бюл. № 7), що включає зчитування даних, виділення даних і їх обробку. Спосіб передбачає попереднє
20 виділення даних двох послідовностей і двоетапне перетворення даних, коли на першому етапі дані другої послідовності підсумовують, а на другому етапі їх масштабують, при цьому коефіцієнт масштабування визначають як відношення суми даних першої послідовності до суми квантованих даних другої послідовності.

Недоліком даного способу є низька оперативність обробки порцій даних системою паралельних обчислень при багатоетапному процесі обробки за рахунок виникнення колізій, що
25 приводить до неможливості їх використання в складних багатоетапних паралельних системах обробки даних.

В основу корисної моделі поставлена задача запобігання виникненню колізій при паралельній обробці цифрових даних шляхом використання системи повідомлень стану та повідомлень-запитів.

30 Поставлена задача вирішується тим, що у способі паралельної обробки цифрових даних із запобіганням виникненню колізій, що містить операції зчитування, обробку даних, згідно з корисною моделлю, що здійснюють безпомилкову паралельну багатоетапну обробку порцій та серій порцій даних із формуванням повідомлень-запитів та стану у відповідності до обробника даних (екземпляра програми).

35 Паралельна обробка цифрових даних із запобіганням виникненню колізій дещо збільшує тривалість процесу обробки даних за умови незначної кількості даних при наявності малої кількості обробників (але не менше двох). Однак, у разі виникнення хоча б однієї колізії часові затрати на повторну обробку порції даних більші за часові витрати обробки цифрових даних із запобіганням виникненню колізій.

40 Крім того, згідно з корисною моделлю, загальний час обробки порцій даних відповідає кількості порцій даних поділеної на кількість обробників обчислювальної системи.

Спосіб паралельної обробки цифрових даних із запобіганням виникненню колізій працює в децентралізованих обчислювальних системах.

45 Спосіб паралельної обробки цифрових даних із запобіганням виникненню колізій працює в централізованих обчислювальних системах.

Заявлений спосіб здійснюється таким чином.

Суть корисної моделі пояснюється кресленнями, де Фіг. 1 - узагальнена блок-схема роботи способу, Фіг. 2 - Фіг. 6 - детальні блок-схеми способу.

На нульовому етапі обробки, що функціонує на обчислювальній станції, екземпляр програми
50 шукає у вхідному буфері чергову порцію даних серії. При відсутності у вхідному буфері порції даних, екземпляр програми виконує пошуковий цикл повідомлень стану по всім сховищам порцій даних на всіх етапах обробки. При виявленні екземпляром програми у сховищі порцій даних повідомлення стану, яке дозволяє проводити обробку серій (повідомлення стану про сформовану і готову до обробки серію порцій даних), він шукає повідомлення-запит на обробку
55 цієї серії порцій даних. У разі відсутності повідомлення-запиту серії порцій даних екземпляр створює в черзі повідомлень-запитів відповідне повідомлення-запит. При наявності повідомлення-запиту цикл починається спочатку.

60 При виявленні екземпляром програми повідомлення стану, яке не дозволяє проводити обробку серій (до таких повідомлень відносяться: повідомлення стану про формування серії порцій даних; повідомлення про процес обробки серії порцій даних; повідомлення про

завершення обробки серії порції даних), він продовжує пошук у сховищах порцій даних. При відсутності завдань (у всіх сховищах порцій даних серії формуються, обробляються або вже оброблені) екземпляр програми надсилає повідомлення програмі про перехід в режим роботи "Очікування". При надходженні повідомлення від програми про вихід з режиму "Очікування" до закінчення заданого часу, екземпляр програми переходить до початку загального циклу. У разі відсутності повідомлення від програми про вихід із режиму "Очікування" і після закінчення заданого часу, екземпляр програми завершує роботу.

При наявності у вхідному буфері порції даних екземпляр програми шукає в черзі повідомлень-запитів повідомлення-запит про ці порції даних. При наявності повідомлення-запиту про порції даних, екземпляр програми переходить до початку загального циклу, а у разі відсутності повідомлення-запиту про порції даних, екземпляр програми створює в черзі повідомлень-запитів відповідне повідомлення-запит.

Після цього екземпляр програми очікує заданий час. Потім він шукає в черзі повідомлень-запитів повідомлення-запит про порції даних.

При виявленні екземпляром програми декількох повідомлень-запитів, дана порція даних буде переміщена цим екземпляром програми за умови, що номер її повідомлення-запиту менше номерів інших повідомлень-запитів. При тотожності номерів повідомлень-запитів, порція даних буде переміщена цим екземпляром програми за умови, що номер обчислювальної станції, на якій він функціонує, менше номерів інших обчислювальних станцій. Решта екземплярів програми переходять до початку циклу.

Після цього екземпляр програми переміщує порцію даних серії з вхідного буфера в сховище порцій даних для першого етапу обробки. Екземпляр програми створює повідомлення стану про формування серії порцій даних з метою запобігання обробки несформованої серії іншим екземпляром програми. Екземпляр програми перевіряє кожну вхідну порцію даних на предмет виявлення останньої із серії. При виявленні чергової (не останньої) порції даних із серії, екземпляр програми переходить до початку циклу. При виявленні останньої порції даних із серії, або, якщо серія містить єдину порцію даних, екземпляр програми створює повідомлення стану про те, що серія порцій даних сформована. Потім він видаляє повідомлення стану про формування серії порцій даних.

Екземпляр програми шукає у черзі повідомлень-запитів повідомлення-запит про серію порцій даних. При наявності повідомлення-запиту про серію порцій даних, екземпляр програми переходить до початку загального циклу. При відсутності повідомлення-запиту, екземпляр програми створює в черзі повідомлень-запитів відповідне повідомлення-запит.

Функціонуючий на обчислювальній станції, екземпляр програми очікує наперед заданий час. Потім він шукає в черзі повідомлень-запитів повідомлення-запит про серію порцій даних. При виявленні екземпляром програми декількох повідомлень-запитів, дана серія порцій даних буде оброблятися цим екземпляром програми за умови, що номер її повідомлення-запиту менше номерів інших повідомлень-запитів, а при тотожності повідомлень-запитів, серія порцій даних буде оброблятися цим екземпляром програми за умови, що номер обчислювальної станції, на якій він функціонує, менше номерів інших обчислювальних станцій. Решта екземплярів програми, переходять до початку загального циклу.

Екземпляр програми здійснює черговий етап обробки серії порцій даних. Перед початком обробки серії порцій даних екземпляр програми створює в черзі повідомлень стану повідомлення стану про процес обробки даної серії порцій даних. Це запобігає зверненню екземпляра програми з метою обробки цієї ж серії порцій даних, що унеможливорює колізію.

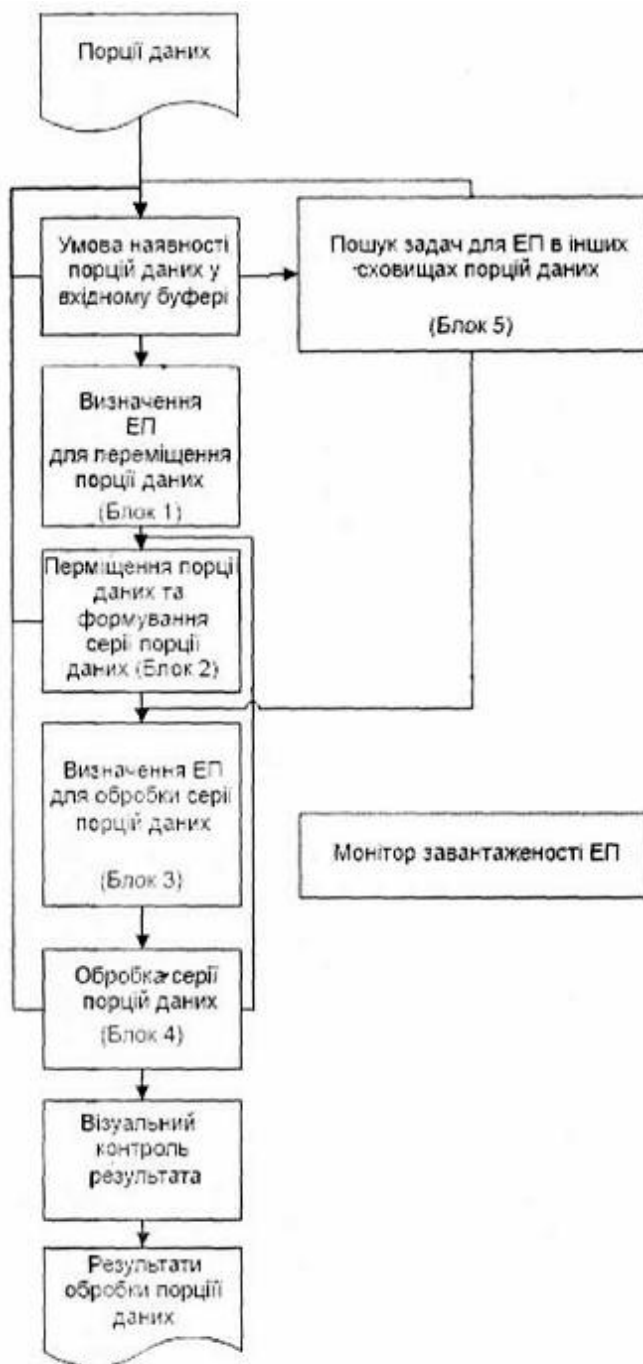
Після закінчення обробки серії порцій даних екземпляр програми створює в черзі повідомлень стану повідомлення про закінчення обробки цієї серії порцій даних. Це запобігає зверненню екземпляра програми з метою повторної обробки цієї ж серії порцій даних. Екземпляр програми видаляє повідомлення стану про процес обробки серії порцій даних.

Після цього екземпляр програми виконує перевірку етапу обробки на предмет визначення його останнього з кількості етапів обробки порцій даних. Якщо це не останній етап обробки, екземпляр програми переходить до пункту формування нової серії для подальшої обробки, а якщо - останній, то екземпляр програми формує звіт-повідомлення про завершення обробки серії порцій даних на останньому етапі і переходить до початку загального циклу.

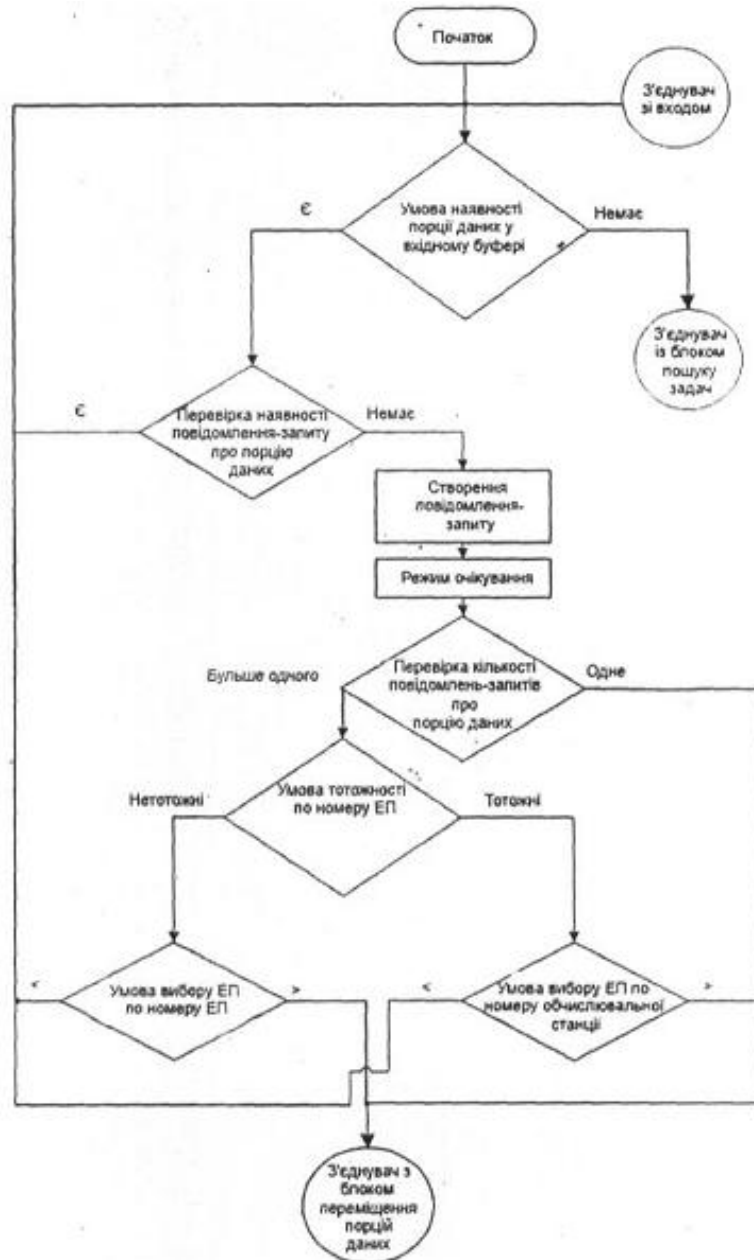
ФОРМУЛА КОРИСНОЇ МОДЕЛІ

Спосіб паралельної обробки цифрових даних із запобіганням виникненню колізій, що містить операції читування, обробку даних, який **відрізняється** тим, що здійснюють безпомилкову

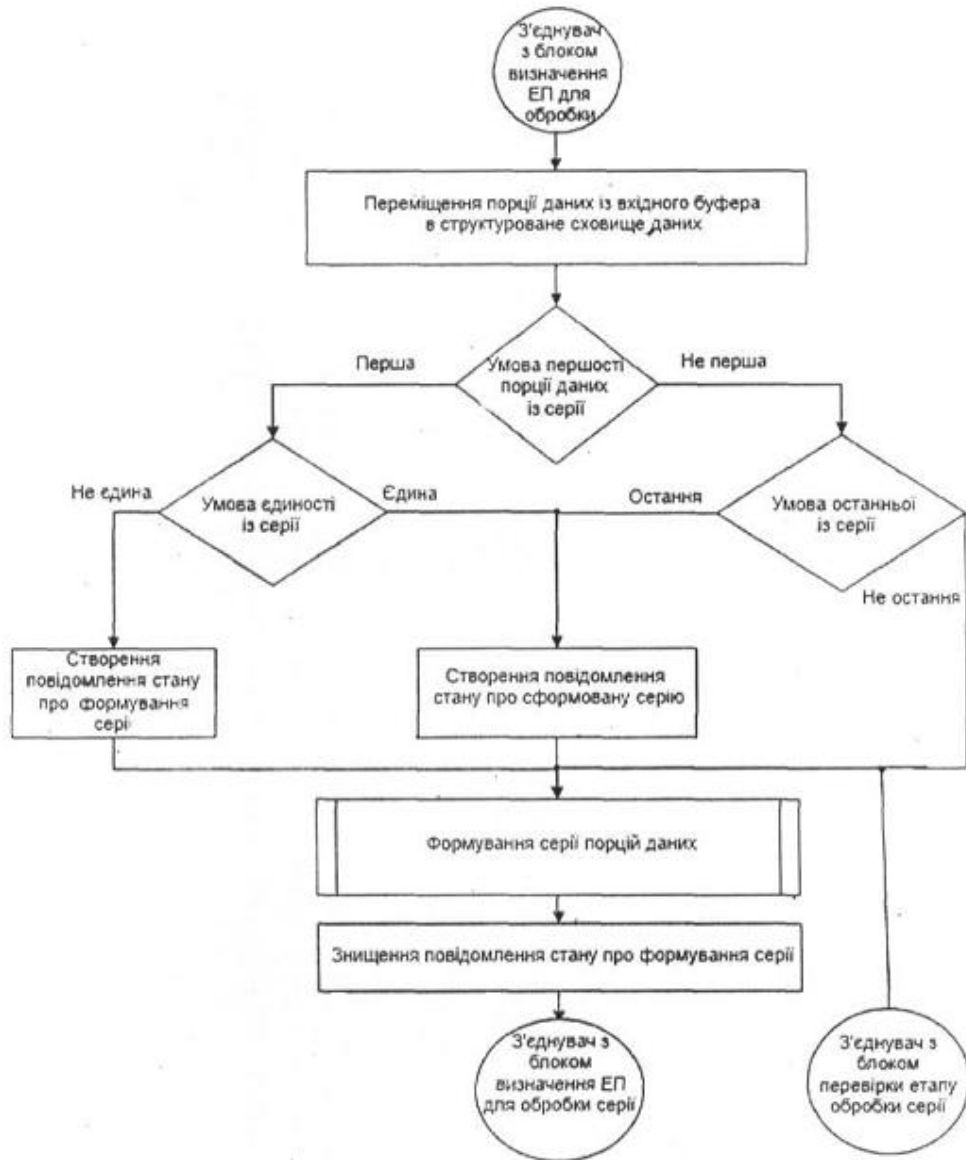
паралельну багатоетапну обробку порцій та серій порцій даних із формуванням повідомлень-запитів та стану у відповідності до обробника даних (екземпляра програми).



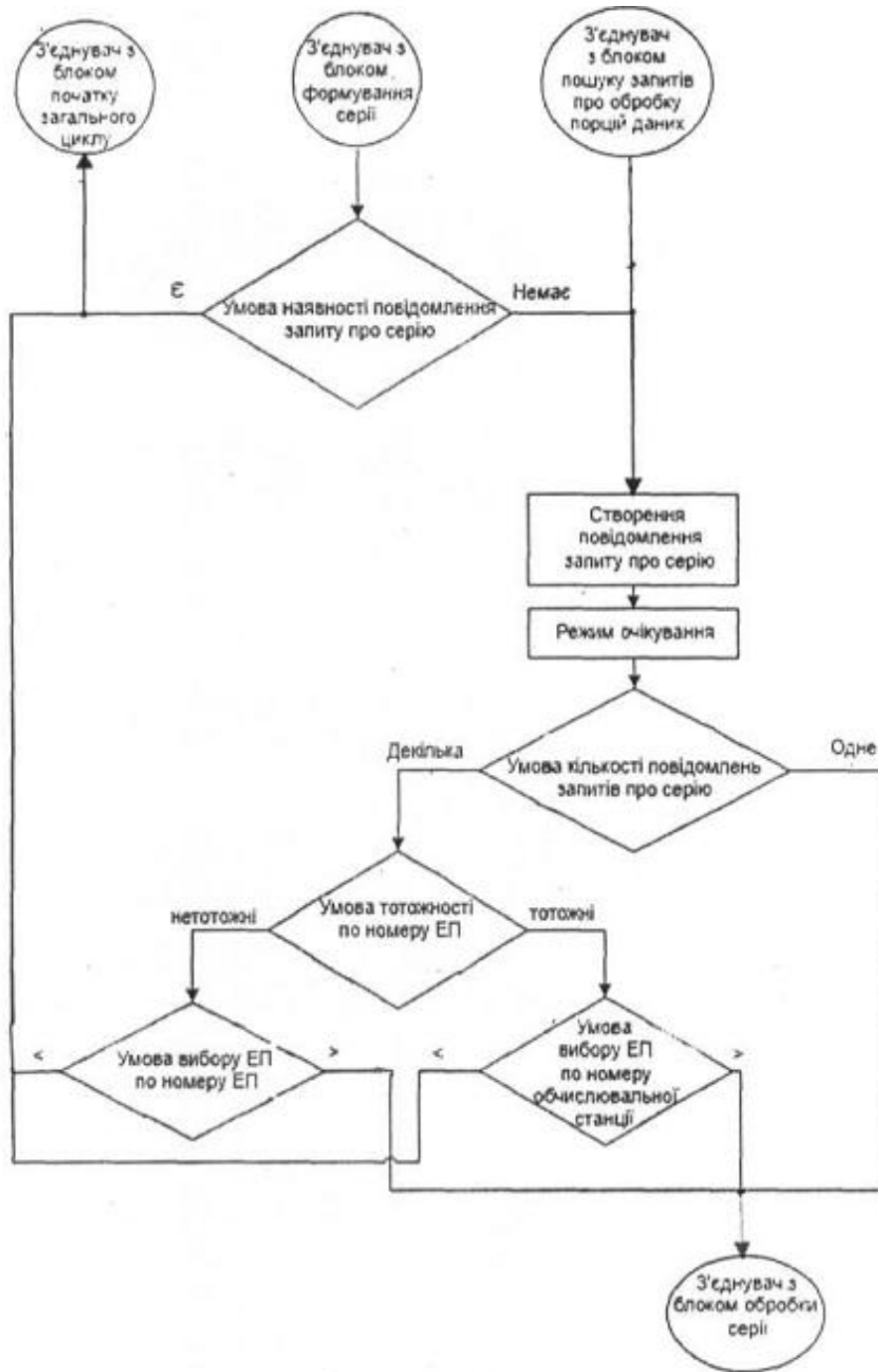
Фіг. 1



Фіг. 2



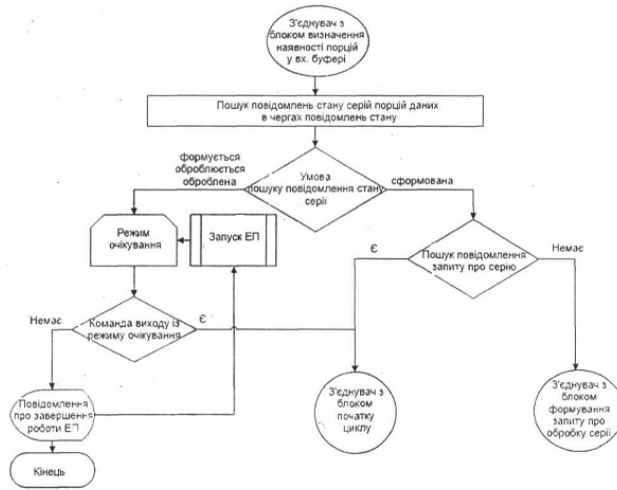
Фіг. 3



Фіг. 4



Фіг. 5



Фіг. 6