



УКРАЇНА

(19) UA

(11) 110055

(13) C2

(51) МПК

G06F 15/18 (2006.01)

G06N 5/02 (2006.01)

G06N 5/04 (2006.01)

ДЕРЖАВНА СЛУЖБА
ІНТЕЛЕКТУАЛЬНОЇ
ВЛАСНОСТІ
УКРАЇНИ

(12) ОПИС ДО ПАТЕНТУ НА ВИНАХІД

(21) Номер заявки: а 2013 14173

(22) Дата подання заявки: 05.12.2013

(24) Дата, з якої є чинними
права на винахід: 10.11.2015

(41) Публікація відомостей
про заявку: 10.06.2015, Бюл.№ 11

(46) Публікація відомостей
про видачу патенту: 10.11.2015, Бюл.№ 21

(72) Винахідник(и):

Кургаєв Олександр Пилипович (UA),
Савченко Іван Васильович (UA)

(73) Власник(и):

ІНСТИТУТ КІБЕРНЕТИКИ ІМ. В.М.
ГЛУШКОВА НАН УКРАЇНИ,
пр. Академіка Глушкова, 40, м. Київ, 03680
(UA)

(56) Перелік документів, взятих до уваги
експертизою:

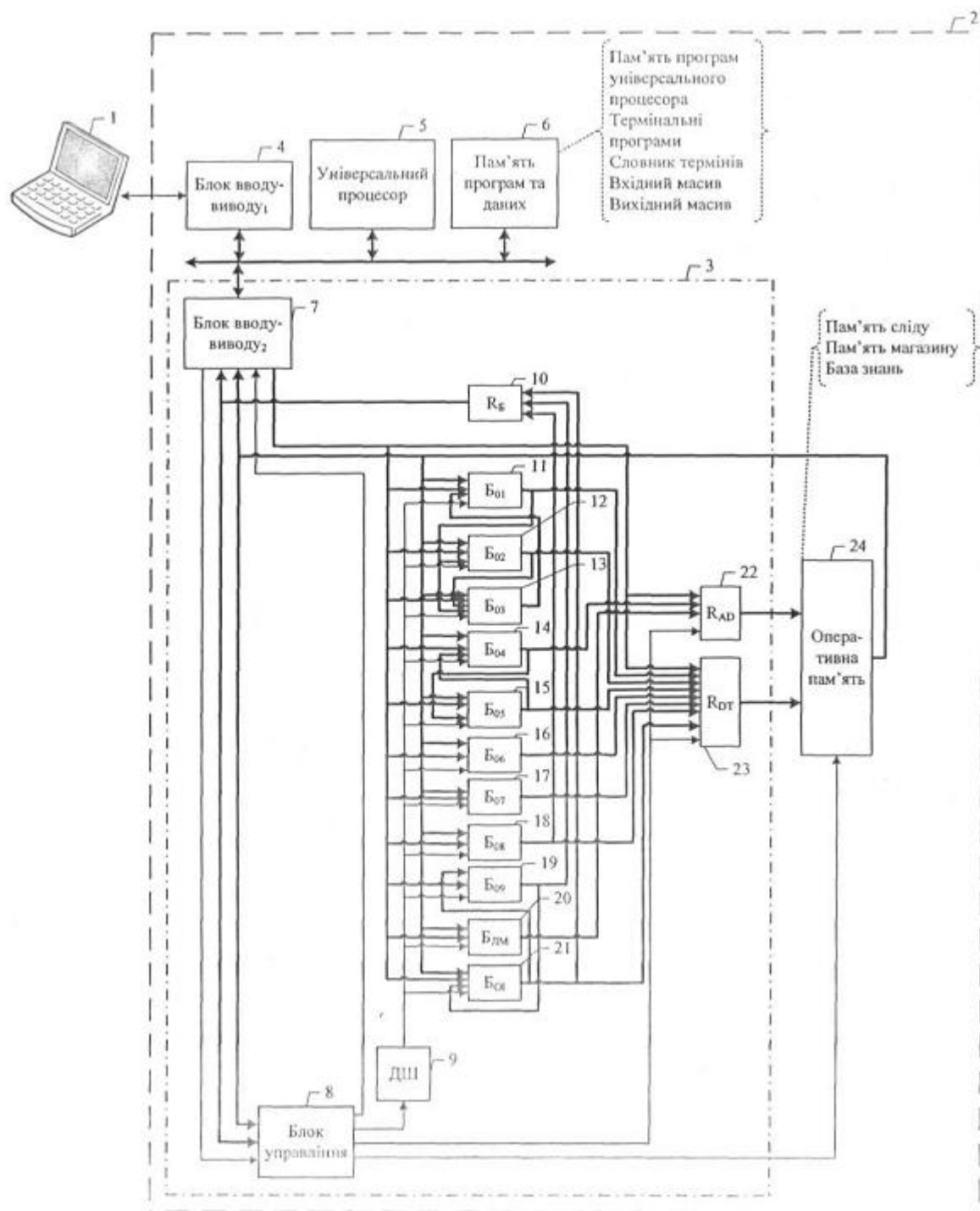
US 5218669 A, 08.06.1993
RU 2092895 C1, 10.10.1997
US 2013117203 A1, 09.05.2013
US 2003217144 A1, 20.11.2003
US 5822743 A, 13.10.1998
US 5253333 A, 12.10.1993
WO 2013130698 A1, 06.09.2013
US 5465319 A, 07.11.1995
GB 2465861 A, 09.06.2010

(54) КОМП'ЮТЕР ОБРОБКИ ЗНАНЬ

(57) Реферат:

Винахід належить до області обчислювальної техніки та може бути використаний при побудові систем, заснованих на знаннях. Комп'ютер обробки знань містить головний комп'ютер, універсальний процесор, оперативну пам'ять, блок управління, блок імені структури, блок номера слова фрейму, блок адреси бази знань, блок координати сліду, блок вершини сліду, блок адреси у вхідному масиві, блок адреси у вихідному масиві, блок ітерації, блок істинності, блок лічильника магазину, блок ознак інтерпретації, буферний регістр, регістр адреси та регістр даних оперативної пам'яті, пам'ять програм та даних, перший та другий блоки вводу-виводу та дешифратор команд. Технічним результатом, що досягається даним винаходом є підвищення ефективності пошуку рішення за рахунок логічного виводу тільки у необхідній частині бази знань, знання, що містяться у базі знань зберігаються у вигляді ієрархічної структури, можливість ефективно зберігати та інтерпретувати ітераційні та рекурсивні структури даних, які містять визначення з використанням відношень альтернативи, послідовності та ітерації.

UA 110055 C2



Фіг. 1

Даний винахід належить до області обчислювальної техніки та може бути використаний при побудові систем, заснованих на знаннях.

Відомий пристрій [SU1455345, 30.01.1989, Устройство для реализации нормальных алгоритмов Маркова // Довгань В.М., Кореневский Н.А., Бойко Ю.Л., Плотников В.В.], який

5 призначено для обробки інформаційних потоків та масивів даних текстового та числового характерів згідно із заданими нормальними алгоритмами та складається із: блока організації підставлення, комутатора, блока пам'яті слова, блока регістрів слова, дешифратора межі слова, блока пошуку входження, блока керування.

Більш швидкодіючим є пристрій [SU1635192, 15.03.91, Устройство для реализации подстановок слов // Довгаль В.М., Корольков О.Ф., Керекеша В.В., Старков Ф.А., Шевелев С.С.], який дозволяє оброблювати дані символьного та числового типів за допомогою нормальних алгоритмів, які складаються виключно із формул підставлення з двобуквеними лівими та правими частинами, при цьому всі ліві частини (антецеденти) формул паралельно порівнюються із двобуквеним початковим словом (ситуацією) за допомогою асоціативного вузла

10 порівняння. Відомий пристрій складається із: блока пам'яті слів, блока пам'яті входжень, блока пам'яті підстановок, вузла зсуву, комутатора, першого вузла контролю оброблюваного слова, вузла порівняння, блока організації підстановки елементів логіки, лічильника та блока керування.

Більш досконалим щодо функціональних можливостей є пристрій [RU2039375, 22.06.1992, Устройство для реализации продукций // Довгаль В.М., Старков Ф.А., Керекеша В.В., Шевелев С.С., Леонов Е.И.], який складається із: блока пам'яті слів, вузла порівняння, блока керування, регістра символу ситуації, регістра маркера ситуації, блока тегів, блока ідентифікації активної продукції та регістра зсуву показника зчитування. Блоки сполучені між собою за допомогою окремих шин та працюють під управлінням блока керування. Серед головних особливостей відомого пристрою (RU2039375, 22.06.1992) слід відмітити те, що він допускає обробку продукції

20 нефіксованої довжини. Головним недоліком даного пристрою є те, що продукції у блоку пам'яті слів зберігаються у вигляді символьного масиву нефіксованої довжини. Це призводить до неефективного використання пам'яті слів та до необхідності по-символьної перевірки в процесі співставлення антецедентів.

Найближчим до запропонованого є пристрій [US Patent, number 5218669 June 8, 1993, VLSI hardware implemented rule-based expert system apparatus and method (апаратно реалізована система обробки продукційних правил) // Hideaki Kobayashi, Masahiro Shindo], обраний за прототип. Пристрій являє собою співпроцесор, що працює під управлінням головного комп'ютера, та містить: блок управління, блок вводу-виводу та оперативну пам'ять.

Пристрій призначений для ефективної обробки продукційних правил та реалізований у формі інтегральної мікросхеми. Блок управління керує роботою всіх модулів пристрою, а також процесом обміну даними між оперативною пам'яттю та блоком вводу-виводу. Оперативна пам'ять містить базу знань у формі системи продукційних правил.

Серед недоліків прототипу можна вказати наступні. 1) Необхідність виконання великої кількості обчислень у процесі пошуку активної продукції при співставленні антецедентів. Це пов'язано із тим, що між продукційними правилами, які містяться в оперативній пам'яті у вигляді команд спеціального формату, відсутній ієрархічний зв'язок. 2) Відсутність механізму пояснення логічного виводу. 3) Відсутність ефективного представлення у пам'яті ітераційних та рекурсивних структур. Спільними ознаками прототипу та пропонованої моделі є головний

40 комп'ютер, оперативна пам'ять, блок управління та перший блок вводу-виводу, перший вхід-вихід якого підключений до системної шини пристрою.

В основу моделі поставлена задача побудови спеціалізованого комп'ютера обробки знань. Будь-яка задача, поставлена у вигляді імені функції та значення її аргументу, вирішується виводом рішення у процесі інтерпретації бази знань.

База знань складається із структури взаємозв'язаних визначень термінів, кожне з яких представляється у формі одного з п'яти фреймів: альтернативи, послідовності, ітерації, текстової константи або терміналу. Усякий фрейм - послідовність слів фіксованої довжини (наприклад, трибайтових), серед яких розрізняється перше слово, а в кожному зі слів - перший байт і наступні. Фрейми текстової константи й терміналу - однослівні, ітерації - двослівні,

50 альтернативи й послідовності - багатослівні. Фрейми визначень всіх термінів всякої бази знань розміщено в оперативній пам'яті бази знань у вигляді ієрархічної структури, що усуває проблему множини конфліктних правил. Такий принцип організації бази знань дозволяє ефективно зберігати та інтерпретувати ітераційні й рекурсивні структури визначень. Ще однією важливою відмінністю комп'ютера обробки знань є те, що структура процесу інтерпретації, процесу виводу рішення зберігається спеціальними засобами ведення сліду.

60

3 метою розширення функціональних можливостей інтерпретації рекурсивних, ітераційних та вкладених структур і формування структури виводу рішення задачі у прототип, що містить головний комп'ютер, блок управління, перший блок вводу-виводу та оперативну пам'ять, у якому головний комп'ютер підключено до першого входу-виходу першого блока вводу-виводу, а перший вихід блока управління підключено до входу управління оперативної пам'яті, введено універсальний процесор, пам'ять програм та даних, другий блок вводу-виводу, дешифратор команд, блок імені структури, блок номера слова фрейму, блок адреси бази знань, блок координати сліду, блок вершини сліду, блок адреси у вхідному масиві, блок адреси у вихідному масиві, блок ітерації, блок істинності, блок лічильника магазину, блок ознак інтерпретації, буферний регістр, регістр адреси оперативної пам'яті та регістр даних оперативної пам'яті, при цьому другий вхід-вихід першого блока вводу-виводу, перший вхід-вихід другого блока вводу-виводу, вхід-вихід універсального процесора, і вхід-вихід пам'яті програм та даних підключені до системної шини пристрою, перший вихід другого блока вводу-виводу підключено до третього входу блока управління, другий вихід якого підключено до третього входу другого блока вводу-виводу, третій вихід блока управління підключено до входу дешифратора команд, вихід якого підключено до четвертих входів блоків імені структури та номера слова фрейму, до п'ятого входу блока адреси бази знань, до четвертих входів блоків координати сліду та вершини сліду, до третіх входів блоків адреси у вхідному та вихідному масиві, до третіх входів блоків ітерації, істинності та лічильника магазину та до четвертого входу блока ознак інтерпретації, четвертий вихід блока управління підключено до четвертого входу регістра адреси оперативної пам'яті та до дев'ятого входу регістра даних оперативної пам'яті, чий вихід підключено до входів адреси і даних оперативної пам'яті, вихід якої підключено до другого входу другого блока вводу-виводу, до першого входу блока управління та до перших входів блоків імені структури, номера слова фрейму, адреси бази знань, координати сліду, вершини сліду, адреси у вхідному масиві, адреси у вихідному масиві, ітерації, лічильника магазину, ознак інтерпретації, а другий вихід другого блока вводу-виводу підключено до других входів блоків імені структури, номера слова фрейму, адреси бази знань, координати сліду, вершини сліду, адреси у вхідному масиві, адреси у вихідному масиві, ітерації, істинності, лічильника магазину, ознак інтерпретації та до перших входів регістру адреси і регістру даних оперативної пам'яті, а вихід блока імені структури підключено до другого входу регістра даних оперативної пам'яті, до третього входу блока номера слова фрейму та до четвертого входу блока адреси бази знань, вихід блока номера слова фрейму підключено до третіх входів регістра даних оперативної пам'яті та блока адреси бази знань, вихід блока адреси бази знань підключено до третього входу блока імені структури, вихід блока координати сліду підключено до другого входу регістра адреси оперативної пам'яті та до третього входу блока вершини сліду, чий вихід підключено до четвертого входу регістра даних оперативної пам'яті та до третього входу блока координати сліду, вихід блока адреси у вхідному масиві підключено до п'ятого входу регістра даних оперативної пам'яті, вихід блока адреси у вихідному масиві підключено до шостого входу регістра даних оперативної пам'яті, вихід блока ітерації підключено до сьомого входу регістра даних оперативної пам'яті та до третього входу буферного регістра, чий другий вхід разом із третім входом блока ознак інтерпретації підключено до виходу блока істинності, вихід блока лічильника магазину підключено до третього входу регістра адреси оперативної пам'яті, вихід блока ознак інтерпретації підключено до перших входів блоків істинності та буферного регістра, та до восьмого входу регістра даних оперативної пам'яті, вихід буферного регістра підключено до першого входу другого блока вводу-виводу та до другого входу блока управління.

Відмінними ознаками запропонованого комп'ютера обробки знань від відомого прототипу є: 1) знання зберігаються у вигляді множини визначень термінів та відокремлено від алгоритму їх обробки; 2) алгоритм роботи пристрою обробки знань за допомогою системи автоматизованого проектування оптимізовано у відповідності з встановленими обмеженнями та апаратно «защитий» у автомат управління по типу Мура; 3) знання, що містяться у базі знань зберігаються у вигляді ієрархічної структури, яка значно підвищує ефективність пошуку рішення за рахунок логічного виводу тільки у необхідній частині бази знань; 4) зберігається структура логічного виводу, яка в подальшому може бути використана для пояснення ходу логічного виводу або породження нової структури даних; 5) надається можливість ефективно зберігати та інтерпретувати ітераційні та рекурсивні структури даних, які містять визначення з використанням відношень кон'юнкції, диз'юнкції та інверсії; 6) модуль розширення функціональності системи реалізовано на базі програмованої логічної інтегральної схеми.

На фіг. 1 наведено структуру комп'ютера обробки знань; на фіг. 2 наведено структуру блока управління; на фіг. 3 наведено структуру блоків B_{01} , B_{03} , B_{06} , B_{07} , B_{LM} , R_{DT} та R_{AD} ; на фіг. 4 наведено структуру блоків B_{02} , B_{04} , B_{05} , B_{08} , B_{09} ; на фіг. 5 наведено структуру блока B_{01} на фіг. 6

наведено структуру запису пам'яті магазина та сліду; на фіг. 7 наведено структуру слова бази знань; на фіг. 8 наведено структуру байта ознак слова фрейму, на фіг. 9 наведено структуру шаблонів фреймів; на фіг. 10 наведено граф алгоритму роботи комп'ютера обробки знань. У табл. 1 наведено реєстровий склад блоків, у табл. 2 наведено опис команд блока управління, у табл. 3 наведено реалізація базових процедур алгоритму роботи системи.

Комп'ютер обробки знань (фіг. 1) складається із: головного комп'ютера 1, модуля розширення функціональності 2, пристрою обробки знань 3, першого блока вводу-виводу 4, універсального процесора 5, пам'яті програм та даних 6, яка складається із пам'яті програм універсального процесора, пам'яті термінальних програм, словника термінів, вхідного та вихідного масиву, другого блока вводу-виводу 7, блока управління 8, дешифратора команд 9, буферного реєстра 10, блока B_{01} імені структури 11, блока B_{02} номера слова фрейму 12, блока B_{03} адреси бази знань 13, блока B_{04} координати сліду 14, блока B_{05} вершини сліду 15, блока B_{06} адреси у вхідному масиві 16, блока B_{07} адреси у вихідному масиві 17, блока B_{08} ітерації 18, блока B_{09} істинності 19, блока $B_{\text{ЛМ}}$ лічильника магазину 20, блока B_{OI} ознак інтерпретації 21, реєстра R_{AD} адреси 22 оперативної пам'яті, реєстра R_{DT} даних 23 оперативної пам'яті та оперативної пам'яті 24.

Головний комп'ютер 1 призначений для постановки завдання, зчитування результатів роботи модуля розширення функціональності 2 та конфігурації роботи модуля 2. Універсальний процесор 5 призначений для керування роботою усіма компонентами із складу модуля 2. Процесор обробки знань 3 розширює набір команд універсального процесора 5 та призначений для швидкої обробки завдання, що надійшло від головного комп'ютера 1. Другий блок вводу-виводу 7 призначений для організації інтерфейсу між головним комп'ютером 1 та модулем 3. Пам'ять 6 призначена для збереження програм універсального процесора, термінальних програм, словника термінів, вхідного масиву та вихідного масиву даних.

Головний комп'ютер 1 через перший блок вводу-виводу 4 підключений разом з універсальним процесором 5, пам'яттю програм та даних 6 та першими входами-виходами другого блока вводу-виводу до системної шини пристрою. Перший вихід другого блока вводу-виводу 7 підключено до третього входу блока управління 8, перший вихід якого підключено до входу управління оперативної пам'яті, а другий вихід підключений до третього входу другого блока вводу-виводу 7. Третій вихід блока управління 8 підключено до входу дешифратора команд 9, вихід якого підключено до четвертих входів блоків імені структури 11 та номера слова фрейму 12, до п'ятого входу блока адреси бази знань 13, до четвертих входів блоків координати сліду 14 та вершини сліду 15, до третіх входів блоків адреси у вхідному 16 та вихідному масиві 17, до третіх входів блоків ітерації 18, істинності 19 та лічильника магазину 20 та до четвертого входу блока ознак інтерпретації 21. Четвертий вихід блока управління 8 підключено до четвертого входу реєстра адреси 22 оперативної пам'яті 24 та до дев'ятого входу реєстра даних 23 оперативної пам'яті 24, вихід якої підключено до другого входу другого блока вводу-виводу 7, до першого входу блока управління 8 та до перших входів блоків імені структури 11, номера слова фрейму 12, адреси бази знань 13, координати сліду 14, вершини сліду 15, адреси у вхідному масиві 16, адреси у вихідному масиві 17, ітерації 18, лічильника магазину 20, ознак інтерпретації 21. Другий вихід другого блока вводу-виводу 7 підключено до других входів блоків імені структури 11, номера слова фрейму 12, адреси бази знань 13, координати сліду 14, вершини сліду 15, адреси у вхідному масиві 16, адреси у вихідному масиві 17, ітерації 18, істинності 19, лічильника магазину 20, ознак інтерпретації 21 та до перших входів реєстра адреси 22 і реєстра даних 23 оперативної пам'яті 24. Вихід блока імені структури 11 підключено до другого входу реєстра даних 23 оперативної пам'яті, до третього входу блока номера слова фрейму 12 та до четвертого входу блока адреси бази знань 13. Вихід блока номера слова фрейму 12 підключено до третіх входів реєстрів даних 23 оперативної пам'яті 24 та блока адреси бази знань 13, вихід якого підключено до третього входу блока імені структури 11. Вихід блока координати сліду 14 підключено до другого входу реєстра адреси 22 оперативної пам'яті 24 та до третього входу блока вершини сліду 15, чий вихід підключено до четвертого входу реєстра даних 23 оперативної пам'яті 24 та до третього входу блока координати сліду 14. Вихід блока 16 адреси у вхідному масиві підключено до п'ятого входу реєстра даних 23 оперативної пам'яті 24. Вихід блока 17 адреси у вихідному масиві підключено до шостого входу реєстра даних 23 оперативної пам'яті 24 та до третього входу буферного реєстра 10, чий другий вхід разом із третім входом блока 21 ознак інтерпретації підключено до виходу блока істинності 19. Вихід блока 20 лічильника магазину підключено до третього входу реєстра адреси 22 оперативної пам'яті 24. Вихід блока 21 ознак інтерпретації підключено до перших входів блоків істинності 19 та буферного реєстра 10, та до восьмого входу реєстра даних 23 оперативної

пам'яті 24. Вихід регістра адреси 22 підключено до першого входу оперативної пам'яті 24. Вихід регістра даних 23 підключено до другого входу оперативної пам'яті 24. Вихід буферного регістра 10 підключено до першого входу другого блока вводу-виводу 7 та до другого входу блока управління 8.

5 Опис блоків із складу пристрою обробки знань 3 наведено у табл. 1.

Зовнішні вхідні дані, що надходять через перший блок вводу-виводу 4, призначені для ініціалізації або модифікації вмісту регістрів блоків модуля 3. Операційна частина пристрою обробки знань виконує команди, набір яких наведено у табл. 2.

Множину команд, наведених у табл. 2, можна розділити на 8 груп:

- 10 а) завантаження даних із R_i до R_j ;
- б) завантаження константи до R_i ;
- в) інкремент даних у R_i та запис їх у R_j ;
- г) декремент даних у R_i та запис їх у R_j ;
- д) завантаження зовнішніх даних у R_i ;
- 15 е) завантаження даних із оперативної пам'яті 24 до R_i ;
- ж) команди для роботи із оперативною пам'яттю 24;
- з) команди запиту на обробку термінальних програм.

Розглянемо принцип роботи комп'ютера обробки знань:

20 1. Ініціалізація пристрою обробки знань. Відбувається ззовні за допомогою блоків вводу-виводу 4 та 7. Залежно від команд, що надійшли від універсального процесора 5, у регістри пристрою 3 та в оперативну пам'ять 24 завантажуються початкові знання та дані. 2. Активізація пристрою обробки знань 3 здійснюється за допомогою передачі блоку 8 управління від універсального процесора 5 спеціальних команд, що містять відповідне завдання щодо використання бази знань.

25 3. Обробка поставленого завдання. Залежно від даних на входах блока управління 8 відбуваються переходи між його станами згідно з графом (фіг. 12), в яких на його виходах формуються команди, які керують виконанням операційною частиною пристрою (блоки 10-23) та оперативною пам'яттю 24 процедур обробки знань.

30 4. Для обробки термінальних програм пристрій 3 за допомогою блока 7 видає універсальному процесору 5 по системній шині пристрою відповідне завдання. Після завершення виконання термінальної програми універсальний процесор 5 надсилає по системній шині пристрою відповідну команду завершення опрацювання термінальної програми та результати її виконання.

35 5. Передача результатів роботи. Після завершення виконання початкового завдання відбувається зчитування по системній шині пристрою даних із оперативної пам'яті 24 та передача їх до головного комп'ютера 1.

40 Операційна пам'ять 24 розміщена на кристалі програмованої логічної інтегральної схеми. Головною її функцією є тимчасове (при наявності живлення кристалу) збереження даних, необхідних в процесі роботи пристрою. Для доступу до оперативної пам'яті 24 застосовуються регістр 22 адреси та регістр 23 даних.

45 Блок 8 управління (фіг. 2) складається із трьох модулів: логіка переходів 25, пам'ять станів 26 та вихідна логіка 27. Логіка переходів 25 визначає наступний стан блока управління 8, що є функцією від поточного стану, що фіксується у пам'яті станів 26, та кодів на входах блока управління 8. Команди на виходах блока управління 8 визначаються вихідною логікою 27 та є функцією від поточного стану пам'яті станів 26.

50 Структуру багатовхідних блоків R_B , B_{01} , B_{03} , R_{DT} , BL_{06} , B_{07} , B_{LM} зображено на фіг. 3 у складі відповідних мультиплексора MX_i та регістра R_i . Вхід мультиплексора MX_i з'єднаний із n шинами вхідних даних та шиною команд. На входи регістра R_i надходять дані з виходу MX_i та відповідна команда. Під управлінням відповідної команди відбувається комутація входів мультиплексора і запис даних у регістр R_i .

55 Структуру B_{02} , B_{04} , B_{05} , B_{08} у формі реверсивного блока зображено на фіг. 4. Реверсивний блок складається із мультиплексора MX_i , арифметико-логічного пристрою та відповідного регістра R_i . Входи мультиплексора MX_i з'єднані із n шинами вхідних даних, виходами арифметико-логічного пристрою та шиною команд. При надходженні відповідної команди відбувається комутація входів мультиплексора MX_i і запис даних у регістр R_i .

Структуру блока інверсії B_{09} зображено на фіг. 5, який складається із мультиплексора MX_i , арифметико-логічного пристрою, інвертора та регістра R_i . Інформаційні входи мультиплексора MX_i з'єднані із шинами вхідних даних, інверсії вхідних даних та виходами арифметико-логічного пристрою, а виходи підключено до входів регістра R_i . Команди з шини команд управляють

комутацією інформаційних входів мультиплексора MX_i та записом даних з його виходів у регістр R_i .

Структуру блока ознак інтерпретації B_{OI} зображено на фіг. 6, який складається із мультиплексора MX_i , регістра ознак сліду (R_{10}), регістра ознак породження (R_{11}), регістра Inv_1 (R_{12}), регістра Inv_2 (R_{13}), регістра типу фрейму (R_{14}), регістра ознаки останнього елемента (R_{15}). Входи мультиплексора MX_i з'єднані із двома шинами вхідних даних та шиною команд. Входи регістрів R_i ($R_{10} - R_{15}$) з'єднані із виходами MX_i та шиною команд. При надходженні відповідної команди відбувається комутація вхідних даних мультиплексора MX_i та їх запис у відповідний регістр R_i .

В оперативній пам'яті 24 розрізняються: пам'ять знань (ПЗ), пам'ять сліду (ПС) та пам'ять магазину (ПМ), які розташовані в ній послідовно.

База знань (БЗ) представляється у формі інформаційної структури множини визначень, вузлами якої є поняття (термінальні - константи чи іменовані процедури, або не термінальні), а дугами - відношення альтернативи, послідовності чи ітерації, що зв'язують поняття (вузли) визначальної частини всякого визначення. Пам'ять бази знань зберігає знання у машинній формі, що складається із множини фреймів, кожний із яких має структуру масиву слів фіксованої довжини (наприклад, 3 байти). Довжина фрейму залежить від типу відношення, яке він описує, та від числа понять у визначальній частині і може складати від одного до деякого скінченного числа слів.

Пам'яті сліду та магазину зберігають (у формі фіг. 7) оперативну інформацію процесу інтерпретації бази знань. Пам'ять сліду зберігає структуру виводу рішення задачі, яка може бути використана для пояснення процесу рішення задачі або при породженні наслідку. Пам'ять магазину зберігає проміжні дані процесу виводу рішення. Якщо на деякому етапі процесу інтерпретації обраний шлях виявився хибним, то відбувається відновлення необхідного вмісту регістрів другої робочої пам'яті зчитуванням даних із пам'яті магазину.

На фіг. 8 наведено структуру слів бази знань, із яких складаються фрейми. Кожне слово бази знань містить 24-біти: байт ознак інтерпретації та інформаційне поле. Інформаційне поле залежно від типу поточного фрейму містить: фізичну адресу, що посилається на інший фрейм; фізичну адресу, що посилається на текстову константу; номер (ім'я) термінальної програми.

На фіг. 9 наведено структуру байта ознак. Дане поле містить інформацію про тип фрейму та ознаки його інтерпретації.

Розглянемо побудову фреймів детальніше (фіг. 10). Кожен фрейм містить голову та елементи, пов'язані одним відношенням: альтернативи, послідовності чи ітерації. Голова та будь-який з елементів займають декілька послідовно розташованих комірок пам'яті, елементи одного фрейму також розташовуються послідовно, причому голова фрейму розташована першою в даній послідовності. Розмір фреймів (число елементів в одному фреймі) альтернативи чи послідовності може бути довільним, але скінченим. Фрейм ітерації складається із двох компонент: голови та посилання на ітерований елемент, яким є фрейм деякого довільного поняття, за виключенням визначеного у формі ітерації. Цим забезпечується зв'язність різних фреймів у єдину багаторазово вкладену структуру, що може містити рекурсивні конструкції. Складність опису (потужність множини понять, структура їхнього взаємозв'язку) може бути довільною та обмежується ресурсами конкретної реалізації (об'ємом оперативної пам'яті, об'ємом зовнішньої пам'яті, розрядністю вузлів та шин).

Ті поняття бази знань, які не визначені через поняття бази знань, є термінальними та містять посилання на константи або термінальні програми, що у подальшому виконуються продукційним процесором. Описана таким чином інформаційна структура прикладної області є базою знань прикладної області.

На фіг. 12 у формі орієнтованого циклічного графа наведено алгоритм роботи системи обробки знань. При виконанні цього алгоритму пристрій виконує набір процедур, зміст яких визначено у табл. 3. У табл. як змінні R_x та R_y можуть бути застосовані регістри $R_{01} - R_{09}$, R_{01} , $R_{ПМ}$.

Пристрій працює наступним чином, безперервно виконуючи алгоритм фіг. 12 від початкової команди «старт» аж до зупинки тактового сигналу. В опису алгоритму використано наступні позначення: ПОЗ - пристрій обробки знань, УП - універсальний процесор обробки процедур, БЗ - база знань, ПМ - пам'ять магазину, ПС - пам'ять сліду, ІНР - пам'ять вхідного масиву, ОУТ - пам'ять вихідного масиву, ПП - пам'ять програм.

1. Початкова установка:

- Завантаження БЗ;

- Завантаження бібліотек термінальних програм в ПП УП або формування (конфігурування)

структури УП;

- Завантаження даних (аргументу обчислювальної функції) в масиви INP і OUT пам'яті даних УП;
- Установка регістрів у початковий стан, зокрема регістр $R_{14}:=00$.
- Перехід в стан 2.
- 5 2. Прийом завдання пристроєм ПОЗ:
 - Якщо отримано завдання, то:
 - прийом від УП адрес даних в R_6 , R_7 та ознак інтерпретації (режиму інтерпретації в R_{10} , R_{11} , значення інверсії \neg 2 в R_{13} та в $R_{15}:=X$);
 - $R_1:=$ адреса БЗ;
 - 10 - запис значень регістрів $[R_6]$, $[R_7]$ в ПМ;
 - запис ознак інтерпретації (R_{14} - тип фрейму=00, режим інтерпретації - $R_{10}=0/1$, $R_{11}=0/1$, R_{12} - $\text{Інв_1}=X$, R_{13} - $\text{Інв_2}=0/1$, $R_{15}=X$) в ПМ;
 - перехід в стан 3.
 - Якщо отримано значення істинності, то зберегти його в R_9 та перейти в стан 21;
 - 15 - Інакше, очікування завдання поверненням до пункту 2.
- 3. Аналіз завдання і режиму його інтерпретації ($R_{14}+R_{10}+R_{11}$):
 - $R_2:= [R_1]$ (вміст R_1 записати в R_2);
 - $R_3:= [R_1]$ (вміст R_1 записати в R_3);
 - Читання першого слова фрейму, $R_{10}':=$ [поле ОС], $R_{14}:=$ [поле «тип фрейму»], $R_{12}:=$ [поле
 - 20 \neg 1].
 - Аналіз R_{14} , R_{10} та R_{11} :
 - Якщо $R_{14}=(00)\&(R_{10}=0/1)\&(R_{11}=0/1)$, то переходимо до п. 4 (термінали),
 - Якщо $(R_{14}=01/10)\&R_{10}=0\&R_{11}=0$, то перехід до п. 5 (розпізнавання альтернативи),
 - Якщо $R_{14}=01\&R_{10}=0\&R_{11}=1$, то перехід до п. 6 (породження альтернативи),
 - 25 Якщо $R_{14}=01\&R_{10}=1\&R_{11}=0$, то перехід до п. 8 (розпізнавання альтернативи із слідом),
 - Якщо $R_{14}=01\&R_{10}=1\&R_{11}=1$, то перехід до п. 9 (породження альтернативи по сліді),
 - Якщо $R_{14}=10\&R_{10}=0\&R_{11}=1$, то перехід до п. 10 (породження послідовності),
 - Якщо $R_{14}=10\&R_{10}=1\&R_{11}=1$, то перехід до п. 12 (породження послідовності по сліді),
 - Якщо $R_{14}=10\&R_{10}=1\&R_{11}=0$, то перехід до п. 13 (розпізнавання послідовності із слідом),
 - 30 Якщо $R_{14}=11\&R_{10}=0\&R_{11}=0$, то перехід до п. 15 (розпізнавання ітерації),
 - Якщо $R_{14}=11\&R_{10}=0\&R_{11}=1$, то перехід до п. 17 (породження ітерації),
 - Якщо $R_{14}=11\&R_{10}=1\&R_{11}=0$, то перехід до п. 18 (розпізнавання ітерації із слідом),
 - Якщо $R_{14}=11\&R_{10}=1\&R_{11}=1$, то перехід до п. 20 (породження ітерації по сліді).
 - 4. Передача термінального завдання процесору ПОД:
 - 35 - $R_1:=$ [перший та другий байти першого слова фрейму];
 - Запис R_6 , R_7 та ознак інтерпретації (тип фрейму - термінал; поточний режим інтерпретації - R_{10} , R_{11} ; інверсія із завдання - \neg 1) в ПМ;
 - Передача R_1 , R_6 , R_7 та ознак інтерпретації процесору ПОД;
 - Перехід в стан 2.
 - 40 5. Розпізнавання альтернативи:
 - $R_2:= [R_2] + 1$;
 - $R_3:= [R_2]$;
 - Читання другого слова фрейму. $R_{13}:=$ [поле \neg 2], $R_{15}:=$ [поле ОЕ], $R_{10}':=$ [значення (0/1) ознаки розпізнавання із слідом] або $R_{11}':=$ [значення (0/1) ознаки породження], $R_1:=$ [перший та
 - 45 другий байти другого слова фрейму].
 - Запис значень регістрів $[R_6]$, $[R_7]$, $[R_2]$ в ПМ. Запис ознак інтерпретації (R_{14} - тип фрейму альтернативи та R_{12} - Інв_1 з першого слова фрейму; R_{13} - Інв_2 , R_{15} - ОЕ з другого слова фрейму та поточний режим інтерпретації - $R_{10}=0$, $R_{11}=0$) в ПМ;
 - $R_{10}:= R_{10}'$; $R_{11}:= R_{11}'$;
 - 50 - Перехід в стан 3.
 - 6. Породження альтернативи:
 - $R_{10}:= R_{10}'$;
 - Аналіз R_{10} (біту «ознака сліді», взятого з першого слова фрейму): якщо 0, то перехід в стан 21, інакше - в 7.
 - 55 7. Встати на слід:
 - $RC_4:=$ [перший та другий байти першого слова фрейму];
 - $RC_8:=$ [ПС за адресою RC_4];
 - $R_3:= [R_2] + [RC_8]$;
 - Читання вибраного слова фрейму. $R_{13}:=$ [поле \neg 2], $R_{15}:=$ [поле ОЕ], $R_1:=$ [перший та
 - 60 другий байти обраного слова фрейму].

- Запис [R6], [R7] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму альтернативи та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2, R15 - ОЕ з вибраного слова фрейму та поточний режим інтерпретації - R10=0/1, R11=1) в ПМ;
 - Перехід в стан 3.

5 8. Розпізнавання альтернативи із слідом:
 - RC8:= 1;
 - RC4:= [R5] + 1;
 - Запис [RC8] в ПС за адресою [RC4];
 - R3:= [R2] + [RC8];

10 - Читання другого слова фрейму. R13:= [поле 2], R15:= [поле ОЕ],
 - R1:= [перший та другий байти вибраного слова фрейму].
 - Запис [R6], [R7], [R5], [R2] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму альтернативи та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2, R15 - ОЕ з вибраного слова фрейму та поточний режим інтерпретації - R10=1, R11=0) в ПМ;

15 - R5:= [RC4];
 - Перехід в стан 3.

9. Породження альтернативи по сліду:
 - RC4:= [RC4] + 1;
 - RC8:= [ПС] за адресою [RC4];

20 - R3:= [R2] + [RC8];
 - Читання вибраного слова фрейму. R13:= [поле 2], R15:= [поле ОЕ],
 - R1:= [перший та другий байти обраного слова фрейму].
 - Запис [R6], [R7] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму альтернативи та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2, R15 - ОЕ з вибраного слова фрейму та поточний режим інтерпретації - R10=1, R11=1) в ПМ;

25 - Перехід в стан 3.

10. Породження послідовності:
 - R10:= R10';
 - Аналіз R10 (біту «ознака сліду», взятого з першого слова фрейму): якщо 1, то переходимо

30 в стан 11, інакше - в 12.

11. Встати на слід:
 - RC4:= [перший та другий байти першого слова фрейму];
 - Перехід в стан 12.

12. Породження послідовності по сліду або без сліду:

35 - R2:= [R2] + 1;
 - R3:= [R2];
 - Читання другого слова фрейму. R13:= [поле 2], R15:= [поле ОЕ], R1:= [перший та другий байти другого слова фрейму].
 - Запис значень регістрів [R6], [R7], [R2] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму послідовності та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2, R15 - ОЕ з другого слова фрейму та поточний режим інтерпретації - R10=0/1, R11=1) в ПМ;

40 - Перехід в стан 3.

13. Розпізнавання послідовності із слідом:
 - RC8:= 1;
 - RC4:= [R5];

45 - R3:= [R2] + [RC8];
 - Читання другого слова фрейму. R13:= [поле 2], R15:= [поле ОЕ], R1:= [перший та другий байти другого слова фрейму].
 - Запис [R6], [R7], [R5], [R2], [RC8] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму послідовності та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2, R15 - ОЕ з другого слова фрейму та поточний режим інтерпретації - R10=0/1, R11=1) в ПМ;

50 - Перехід в стан 3.

14. Завершити породження ітерації по сліду:
 - RC8:= [RC8] - 1;

55 - R3:= [R2] + 1;
 - Читання другого слова фрейму. R13:= [поле 2], R1:= [перший та другий байти другого слова фрейму].
 - Запис [RC8], [R2] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму ітерації та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2 з другого слова фрейму та поточний режим інтерпретації - R10=0/1, R11=1) в ПМ;

60

- Перехід в стан 3.

15. Розпізнавання ітерації:

- $R3 := [R2] + 1$;

- Читання другого слова фрейму. $R13 := [\text{поле } \lfloor 2]$, $R15 := [\text{поле } OE]$, $R10' := [\text{значення } (0/1)$
5 $\text{ознаки розпізнавання із слідом}]$ або $R11' := [\text{значення } (0/1) \text{ ознаки породження}]$, $R1 := [\text{перший та}$
 $\text{другий байти другого слова фрейму}]$.

- Запис значень регістра $[R2]$ в ПМ. Запис ознак інтерпретації ($R14$ - тип фрейму ітерації та
 $R12$ - Інв_1 з першого слова фрейму; $R13$ - Інв_2 , $R15$ - OE з другого слова фрейму та поточний
режим інтерпретації - $R10=0$, $R11=0$) в ПМ;

10 - $R10 := R10'$; $R11 := R11'$;

- Перехід в стан 3.

16. Встати на слід:

- $RC4 := [\text{другий та третій байти першого слова фрейму}]$;

- $RC8 := [PC]$ по адресу $RC4$;

15 - Якщо лічильник $[RC8] \neq 0$, то перехід в стан 14, інакше - в стан 19.

17. Породження ітерації:

- $R10 := R10'$;

- Аналіз $R10$ (біта «ознака сліду», взятого з першого слова фрейму): якщо 1, то переходимо
в стан 16, інакше - в 19.

20 18. Розпізнавання ітерації із слідом:

- $RC4 := [R5] + 1$;

- $RC8 := 0$;

- Запис $[RC8]$ в ПС по адресу $RC4$;

- $R3 := [R2] + 1$;

25 - Читання другого слова фрейму. $R13 := [\text{поле } \lfloor 2]$, $R1 := [\text{перший та другий байти другого}$
 $\text{слова фрейму}]$.

- Запис $[R5]$, $[R2]$ в ПМ. Запис ознак інтерпретації ($R14$ - тип фрейму ітерації та $R12$ - Інв_1 з
першого слова фрейму; $R13$ - Інв_2 з другого слова фрейму та поточний режим інтерпретації -
 $R10=1$, $R11=0$) в ПМ;

30 - $R5 := [RC4]$;

- Перехід в стан 3.

19. Завершити породження ітерації без сліду:

- $R9 := R12$;

- Перехід в стан 21.

35 20. Породження ітерації по сліду:

- $RC4 := [RC4] + 1$;

- $RC8 := [PC]$ по адресу $RC4$;

- Якщо $[RC8] \neq 0$, то перехід в стан 14, інакше - в стан 19.

21. Прийом і аналіз прапорів:

40 - Ознаки інтерпретації з магазину R_{10} , R_{11} , R_{12} , R_{13} , R_{14} , $R_{15} := [\text{Верхній запис ПМ}]$;

- $R9 := [R9] \oplus [R13]$;

- Аналіз $R14$, $R9$, $R10$ та $R11$:

Якщо $R14=(00) \& R9=1 \& (R10=0/1) \& (R11=0/1)$, то переходимо до п. 22 (прапор лож на
термінал),

45 Якщо $R14=(00) \& R9=0 \& (R10=0/1) \& (R11=0/1)$, то переходимо до п. 23 (прапор істина на
термінал),

Якщо $R14=01 \& R9=0 \& R10=0 \& R11=0$, то перехід до п. 24 (розпізнавання альтернативи після
прапора істина),

Якщо $R14=01 \& R9=0 \& R10=1 \& R11=0$, то перехід до п. 25 (розпізнавання альтернативи із
50 слідом після прапора істина),

Якщо $R14=01 \& R9=0 \& R10=0/1 \& R11=1$, то перехід до п. 27 (породження альтернативи після
прапора істина),

Якщо $R14=01 \& R9=1 \& R10=0 \& R11=0$, то перехід до п. 28 (розпізнавання альтернативи після
прапора лож),

55 Якщо $R14=01 \& R9=1 \& R10=1 \& R11=0$, то перехід до п. 30 (розпізнавання альтернативи із
слідом після прапора лож),

Якщо $R14=01 \& R9=1 \& R10=0/1 \& R11=1$, то перехід до п. 32 (породження альтернативи після
прапора лож),

Якщо $R14=10 \& R9=0 \& R10=0 \& R11=0$, то перехід до п. 34 (розпізнавання послідовності після
60 прапора істина),

Якщо $R14=10 \& R9=0 \& R10=1 \& R11=0$, то перехід до п. 35 (розпізнавання послідовності із слідом після прапора істина),

Якщо $R14=10 \& R9=0 \& R10=0/1 \& R11=1$, то перехід до п. 37 (породження послідовності після прапора істина),

5 Якщо $R14=10 \& R9=1 \& R10=0 \& R11=0$, то перехід до п. 38 (розпізнавання послідовності після прапора лож),

Якщо $R14=10 \& R9=1 \& R10=1 \& R11=0$, то перехід до п. 40 (розпізнавання послідовності із слідом після прапора лож),

10 Якщо $R14=10 \& R9=1 \& R10=0/1 \& R11=1$, то перехід до п. 41 (породження послідовності після прапора лож),

Якщо $R14=11 \& R9=0 \& R10=0 \& R11=0$, то перехід до п. 42 (розпізнавання ітерації після прапора істина),

Якщо $R14=11 \& R9=0 \& R10=1 \& R11=0$, то перехід до п. 43 (розпізнавання ітерації із слідом після прапора істина),

15 Якщо $R14=11 \& R9=0 \& R10=0/1 \& R11=1$, то перехід до п. 44 (породження ітерації після прапора істина),

Якщо $R14=11 \& R9=1 \& R10=0 \& R11=0$, то перехід до п. 46 (розпізнавання ітерації після прапора лож),

20 Якщо $R14=11 \& R9=1 \& R10=1 \& R11=0$, то перехід до п. 47 (розпізнавання ітерації із слідом після прапора лож),

Якщо $R14=11 \& R9=1 \& R10=0/1 \& R11=1$, то перехід до п. 48 (породження ітерації після прапора лож).

22. Відновлення показчиків INP, OUT:

25 - R6:= [R6 із ПМ];

- R7:= [R7 із ПМ];

- Перехід в стан 2.

23. Набуття нових значень показчиків INP, OUT:

- Виштовхнути [R6], [R7] із ПМ;

- Перехід в стан 2.

30 24. Розпізнавання альтернативи після прапора істина:

- Виштовхнути [R2 з ПМ];

- Виштовхнути [R6 та R7] із ПМ.

- $R9 := [R9] \oplus [R12]$;

- Перехід в стан 21.

35 25. Розпізнавання альтернативи із слідом після прапора істина:

- R2:= [R2 із ПМ];

- R3:= [R2];

- RC4:= [R5 із ПМ];

- ОС першого слова фрейму:= 1; поля \lceil 1 та тип фрейму - відновити;

40 - [перший та другий байти першого слова фрейму] := [RC4];

- Виштовхнути [R6 та R7] із ПМ.

- $R9 := [R9] \oplus [R12]$;

- Перехід в стан 21.

26. Вихід з розпізнавання альтернативи:

45 - Виштовхнути [R2 із ПМ];

- $R9 := [R9] \oplus [R12]$;

- Перехід в стан 21.

27. Породження альтернативи по сліду після прапора істина:

- Виштовхнути [R6 та R7] із ПМ.

50 - $R9 := [R9] \oplus [R12]$;

- Перехід в стан 21.

28. Розпізнавання альтернативи після прапора лож:

- Відновлення показчиків INP, OUT: R6:= [R6 з ПМ], R7:= [R7 з ПМ];

- Аналіз значення R15: якщо 1, перехід в стан 26, інакше - в стан 29.

55 29. Розпізнавання нового варіанта альтернативи:

- R2:= [R2 із ПМ] + 1;

- R3:= [R2];

- Читання чергового слова фрейму альтернативи. R13:= [поле \lceil 2], R15:= [поле OE], R10':= [значення (0/1) ознаки розпізнавання із слідом] або R11':= [значення (0/1) ознаки породження], R1:= [перший та другий байти вибраного слова фрейму].

60

- Запис значень регістрів [R6], [R7], [R2] в ПМ Запис ознак інтерпретації (R14 - тип фрейму альтернативи та R12 - Інв_1 з першого слова фрейму - тобто відновлення їхніх значень у магазині; R13 - Інв_2, R15 - OE з чергового слова фрейму та відновити записаний у магазині вхідний режим інтерпретації - R10=0, R11=0) в ПМ;
- 5 - R10:=R10'; R11:=R11';
- Перехід в стан 3.
- 30. Розпізнавання альтернативи із слідом після прапора лож:
- Відновлення показчиків INP, OUT: R6:= [R6 з ПМ], R7:= [R7 з ПМ];
- Аналіз значення R15: якщо 1, перехід в стан 31, інакше - в стан 33.
- 10 31. Вихід із розпізнавання альтернативи із слідом:
- Виштовхнути [R2 із ПМ];
- R5:= [R5 із ПМ];
- R9:= [R9] \oplus [R12];
- Перехід в стан 21.
- 15 32. Породження альтернативи по сліду після прапора лож:
- Відновлення показчиків R6:= [R6 із ПМ], R7:= [R7 із ПМ];
- R9:= [R9] \oplus [R12];
- Перехід в стан 21.
- 33. Розпізнавання із слідом нового варіанта альтернативи:
- 20 - RC4:= [R5 із ПМ] + 1;
- RC8:= [RC8 із ПС] + 1;
- Запис [RC8] в ПС по адресу RC4;
- R2:= [R2 із ПМ];
- R3:= [R2] + [RC8];
- 25 - Читання чергового слова фрейму. R13:= [поле \lceil 2], R15:= [поле OE],
- R1:= [перший та другий байти вибраного слова фрейму].
- Запис [R6], [R7], [R5], [R2] в ПМ - ця запис відновлюється;
- Запис ознак інтерпретації (R14 - тип фрейму альтернативи та R12 - Інв_1 з першого слова фрейму, відновлених з магазину; R13 - Інв_2, R15 - OE з вибраного слова фрейму та поточний режим інтерпретації - R10=1, R11=0) в ПМ;
- 30 - R5:= [RC4];
- Перехід в стан 3.
- 34. Розпізнавання послідовності після прапора істина:
- Аналіз значення R15: якщо 1, перехід в стан 24, інакше - в стан 29.
- 35 35. Розпізнавання послідовності із слідом після прапора істина:
- Аналіз значення R15: якщо 1, перехід в стан 25, інакше - в стан 36.
- 36. Продовжити розпізнавання послідовності із слідом:
- RC8:= [RC8 із ПМ] + 1;
- R3:= [R2] + [RC8];
- 40 - Читання чергового слова фрейму. R13:= [поле \lceil 2], R15:= [поле OE], R1:= [перший та другий байти обраного слова фрейму].
- Запис [R6], [R7], [R5], [R2], [RC8] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму послідовності та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2, R15 - OE з другого слова фрейму та поточний режим інтерпретації - R10=0/1, R11=1) в ПМ;
- 45 - Перехід в стан 3.
- 37. Породження послідовності після прапора істина:
- Аналіз значення R15: якщо 1, перехід в стан 24, інакше - в стан 39.
- 38. Розпізнавання послідовності після прапора лож:
- Відновлення показчиків INP, OUT: R6:= [R6 з ПМ], R7:= [R7 з ПМ];
- 50 - Перехід в стан 26.
- 39. Продовжити породження послідовності:
- R2:= [R2 із ПМ] + 1;
- R3:= [R2];
- Читання чергового слова фрейму послідовності. R13:= [поле \lceil 2], R15:= [поле OE], R1:= [перший та другий байти другого слова фрейму].
- 55 - Запис [R6], [R7], [R2] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму послідовності та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2, R15 - OE з другого слова фрейму та поточний режим інтерпретації - R10=0/1, R11=1) в ПМ;
- Перехід в стан 3.
- 60 40. Розпізнавання послідовності із слідом після прапора лож:

- Відновлення показчиків: R5:= [R5 із ПМ], R6:= [R6 із ПМ], R7:= [R7 із ПМ];
- R10:= R10'; R11:= R11';
- Перехід в стан 26.
- 41. Породження послідовності після прапора лож:
- 5 - Відновлення показчиків INP, OUT: R6:= [R6 із ПМ], R7:= [R7 із ПМ];
- R10:=R10'; R11:= 11';
- Виштовхнути [R2 із ПМ];
- R9:= [R9] \oplus [R12];
- Перехід в стан 21.
- 10 42. Розпізнавання ітерації після прапора істина:
- R3:= [R2 із ПМ] + 1;
- Читання другого слова фрейму. R13:= [поле \lceil 2], R15:= [поле ОЕ], R10':= [значення (0/1) ознаки розпізнавання із слідом] або R11':= [значення (0/1) ознаки породження], R1:= [перший та другий байти другого слова фрейму].
- 15 - Запис значення регістру [R2] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму ітерації та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2, R15 - ОЕ з другого слова фрейму та поточний режим інтерпретації - R10=0, R11=0) в ПМ; - тобто тут просто збільшення лічильника магазину без фактичного запису (запис зберігається)
- R10:= R10'; R11:= R11';
- 20 - Перехід в стан 3.
- 43. Розпізнавання ітерації із слідом після прапора істина:
- RC4:= [R5 із ПМ] + 1;
- RC8:= [RC8 із ПС по адресу RC4] + 1;
- Запис [RC8] в ПС по адресу RC4;
- 25 - R3:= [R2 із ПМ] + 1;
- Читання другого слова фрейму. R13:= [поле \lceil 2], R1:= [другий та третій байти другого слова фрейму].
- Запис [R5], [R2] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму ітерації та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2 з другого слова фрейму та поточний режим інтерпретації - R10=1, R11=0) в ПМ; - тобто тут просто збільшення лічильника магазину без фактичного запису (запис зберігається)
- 30 - Перехід в стан 3.
- 44. Породження ітерації після прапора істина:
- RC8:= [RC8 із ПМ];
- 35 - Аналіз [RC8]: якщо [RC8] \neq 0, то перехід в стан 45, інакше - в стан 46.
- 45. Продовжити породження ітерації після прапора істина:
- RC8:= [RC8] - 1;
- R3:= [R2 із ПМ] + 1;
- Читання другого слова фрейму. R13:= [поле \lceil 2], R1:= [перший та другий байти другого слова фрейму].
- 40 - Запис [RC8], [R2] в ПМ. Запис ознак інтерпретації (R14 - тип фрейму ітерації та R12 - Інв_1 з першого слова фрейму; R13 - Інв_2 з другого слова фрейму та поточний режим інтерпретації - R10=0/1, R11=1) в ПМ;
- Перехід в стан 3.
- 45 46. Завершити породження ітерації та розпізнавання ітерації після прапора лож:
- Виштовхнути [R2 із ПМ];
- R9:= [R12];
- Перехід в стан 21.
- 47. Розпізнавання ітерації із слідом після прапора лож:
- 50 - R3:= [R2 із ПМ];
- RC4:= [R5 із ПМ];
- ОС першого слова фрейму:= 1; поля \lceil 1 та тип фрейму - відновити;
- [перший та другий байти першого слова фрейму]:= [RC4];
- R9:= [R12];
- 55 - Перехід в стан 21.
- 48. Породження ітерації після прапора лож:
- Виштовхнути [R2 із ПМ];
- Виштовхнути [RC8 із ПМ];
- R9:= [R12];
- 60 - Перехід в стан 21.

Таблиця 1

Блок	Регістр, який містить блок		Розрядність, біт	Опис регістра
Б ₀₁	R ₀₁		16	Регістр імені структури
Б ₀₂	R ₀₂		16	Регістр номера слова фрейму
Б ₀₃	R ₀₃		16	Регістр адреси бази знань
Б ₀₄	RC ₀₄		16	Регістр координати сліду
Б ₀₅	R ₀₅		16	Регістр вершини сліду
Б ₀₆	R ₀₆		16	Регістр адреси у вхідному масиві
Б ₀₇	R ₀₇		16	Регістр адреси у вихідному масиві
Б ₀₈	RC ₀₈		16	Регістр лічильника ітерації
Б ₀₉	R ₀₉		16	Регістр істинності
Б _{0i}	R _{0i}	R ₁₀	1	Регістр ознаки сліду
		R ₁₁	1	Регістр ознаки породження
		R ₁₂	1	Регістр Інкв_1
		R ₁₃	1	Регістр Інкв_2
		R ₁₄	3	Регістр типу фрейму
		R ₁₅	1	Регістр ознаки останнього елемента
Б _{ЛМ}	R _{ЛМ}		16	Регістр лічильника пам'яті магазину
R _{AD}	R _{AD}		16	Регістр адреси даних в оперативній пам'яті
R _{DT}	R _{DT}		16	Регістр даних в оперативній пам'яті

Таблиця 2

№	Ім'я команди	Опис команди
1	LD $\overline{R_{12}}$, R ₀₉	Завантаження $\overline{R_{12}}$ у R ₀₉
2	LD 0, R ₀₈	Завантаження «0» у R ₀₈
3	LD 1, R ₁₁	Завантаження «1» у R ₁₁
4	LD 1, R ₁₀	Завантаження «1» у R ₁₀
5	LD R ₀₁ , R ₀₂	Завантаження даних із R ₀₁ у R ₀₂
6	LD R ₀₁ , R ₀₃	Завантаження даних із R ₀₁ у R ₀₃
7	LD R ₀₁ , R _{DT}	Завантаження даних із R ₀₁ у R _{DT}
8	LD R ₀₁ +1, R ₀₂	Інкремент даних у R ₀₁ та запис їх у R ₀₂
9	LD R ₀₂ , R ₀₃	Завантаження даних із R ₀₂ у R ₀₃
10	LD R ₀₂ , R _{DT}	Завантаження даних із R ₀₂ у R _{DT}
11	LD R ₀₂ +1, R ₀₂	Інкремент даних у R ₀₂ та запис їх у R ₀₂
12	LD R ₀₄ , R ₀₅	Завантаження даних із R ₀₄ у R ₀₅
13	LD R ₀₄ , R _{RA}	Завантаження даних із R ₀₄ у R _{RA}
14	LD R ₀₄ , R _{WA}	Завантаження даних із R ₀₄ у R _{WA}
15	LD R ₀₄ +1, R ₀₄	Інкремент даних у R ₀₄ та запис їх у R ₀₄
16	LD R ₀₅ , R ₀₄	Завантаження даних із R ₀₅ у R ₀₄
17	LD R ₀₅ , R _{DT}	Завантаження даних із R ₀₅ у R _{DT}
18	LD R ₀₅ +1, R ₀₄	Інкремент даних у R ₀₅ та запис їх у R ₀₄
19	LD R ₀₆ , R _{DT}	Завантаження даних із R ₀₆ у R _{DT}
20	LD R ₀₇ , R _{DT}	Завантаження даних із R ₀₇ у R _{DT}
21	LD R ₀₈ , R _{DT}	Завантаження даних із R ₀₈ у R _{DT}
22	LD R ₀₈ -1, R ₀₈	Декремент даних у R ₀₈ та запис їх у R ₀₈
23	LD R ₀₉ \oplus R ₁₂ , R ₀₉	Завантаження R ₀₉ \oplus R ₁₂ у R ₀₉
24	LD R ₀₉ \oplus R ₁₃ , R ₀₉	Завантаження R ₀₉ \oplus R ₁₃ у R ₀₉
25	LD R _{0i} , R _{DT}	Завантаження даних із R _{0i} у R _{DT}
26	LD R _{ЛМ} , R _{WA}	Завантаження даних із R _{ЛМ} у R _{WA}
27	LD R _{ЛМ} + 1; R _{ЛМ}	Інкремент даних у R _{ЛМ} та запис їх у R _{ЛМ}
28	LD R _{ЛМ} - 1, R _{ЛМ}	Декремент даних у R _{ЛМ} та запис їх у R _{ЛМ}
29	LD DATA _{EXT} , R ₀₁	Завантаження зовнішніх даних у R ₀₁
30	LD DATA _{EXT} , R ₀₂	Завантаження зовнішніх даних у R ₀₂
31	LD DATA _{EXT} , R ₀₃	Завантаження зовнішніх даних у R ₀₃
32	LD DATA _{EXT} , R ₀₄	Завантаження зовнішніх даних у R ₀₄
33	LD DATA _{EXT} , R ₀₅	Завантаження зовнішніх даних у R ₀₅
34	LD DATA _{EXT} , R ₀₆	Завантаження зовнішніх даних у R ₀₆

Продовження табл. 2

№	Ім'я команди	Опис команди
35	LD DATA _{EXT} , R ₀₇	Завантаження зовнішніх даних у R ₀₇
36	LD DATA _{EXT} , R ₀₈	Завантаження зовнішніх даних у R ₀₈
37	LD DATA _{EXT} , R ₀₉	Завантаження зовнішніх даних у R ₀₉
38	LD DATA _{EXT} , R _{DT}	Завантаження зовнішніх даних у R _{DT}
39	LD DATA _{EXT} , R _{RA}	Завантаження зовнішніх даних у R _{RA}
40	LD DATA _{EXT} , R _{WA}	Завантаження зовнішніх даних у R _{WA}
41	LD DATA _{EXT} , R _{OI}	Завантаження зовнішніх даних у R _{OI}
42	LD DATA _{EXT} , R _{LM}	Завантаження зовнішніх даних у R _{LM}
43	LD DATA _{RAM} , R ₀₁	Завантаження даних із оперативної пам'яті у R ₀₁
44	LD DATA _{RAM} , R ₀₂	Завантаження даних із оперативної пам'яті у R ₀₂
45	LD DATA _{RAM} , R ₀₄	Завантаження даних із оперативної пам'яті у R ₀₄
46	LD DATA _{RAM} , R ₀₅	Завантаження даних із оперативної пам'яті у R ₀₅
47	LD DATA _{RAM} , R ₀₆	Завантаження даних із оперативної пам'яті у R ₀₆
48	LD DATA _{RAM} , R ₀₇	Завантаження даних із оперативної пам'яті у R ₀₇
49	LD DATA _{RAM} , R ₀₈	Завантаження даних із оперативної пам'яті у R ₀₈
50	LD DATA _{RAM} , R _{OI}	Завантаження даних із оперативної пам'яті у R _{OI}
51	RD RAM	Зчитування даних із оперативної пам'яті
52	REQ_TER_PROGR	Запит ресурсів універсального процесора на опрацювання термінальної програми
53	WR RAM	Запис даних у оперативну пам'ять
54	LD R ₀₈ , R _Б	Завантаження даних із R ₀₈ у R _Б
55	LD R ₀₉ , R _Б	Завантаження даних із R ₀₉ у R _Б
56	LD R _{OI} , R _Б	Завантаження даних із R _{OI} у R _Б

Таблиця 3

№	Базова процедура	Реалізація базових процедур
1	Вміст R _x записати в R _y	LD R _x , R _y
2	Зчитування слова із оперативної пам'яті по адресу, який міститься в R _y , та запис в R _x	LD R _y , R _{RA} RD RAM LD DATA _{RAM} , R _x
3	Запис даних, що знаходяться в регістрі R _x в оперативну пам'ять, по адресу, що знаходиться в R _y	LD R _y , R _{WA} LD R _x , R _{DT} WR RAM
4	Інкремент даних у R _x та їх запис у R _y	LD R _x + 1, R _y
5	Декремент даних у R _x та їх запис у R _y	LD R _x - 1, R _y
6	Запис «0» у R _x	LD 0, R _x
7	Запис «1» у R _x	LD 1, R _x
8	Інвертування R _x та запис отриманих даних у R _y	LD $\overline{R_x}$, R _y
9	Завантаження зовнішніх даних у R _x	LD DATA _{EXT} , R _x
10	Зчитування першого слова фрейму	LD R ₀₂ , R _{RA} RD RAM
11	Аналіз R _x	LD R _x , R _Б
12	Передача R _x , до універсального процесора	LD R _x , R _Б REQ_TER_PROGR

5

ФОРМУЛА ВИНАХОДУ

10

Комп'ютер обробки знань, що містить головний комп'ютер, блок управління, перший блок вводу-виводу та оперативну пам'ять, причому головний комп'ютер підключено до першого входу-виходу першого блока вводу-виводу, а перший вихід блока управління підключено до входу управління оперативної пам'яті, який **відрізняється** тим, що в нього введено універсальний процесор, пам'ять програм та даних, другий блок вводу-виводу, дешифратор команд, блок імені структури, блок номера слова фрейму, блок адреси бази знань, блок координати сліду, блок вершини сліду, блок адреси у вхідному масиві, блок адреси у вихідному масиві, блок ітерації, блок істинності, блок лічильника магазину, блок ознак інтерпретації, буферний регістр, регістр

адреси оперативної пам'яті та реєстр даних оперативної пам'яті, при цьому другий вхід-вихід першого блока вводу-виводу, перший вхід-вихід другого блока вводу-виводу, вхід-вихід універсального процесора, і вхід-вихід пам'яті програм та даних підключені до системної шини пристрою, перший вихід другого блока вводу-виводу підключено до третього входу блока управління, другий вихід якого підключено до третього входу другого блока вводу-виводу, третій вихід блока управління підключено до входу дешифратора команд, вихід якого підключено до четвертих входів блоків імені структури та номера слова фрейму, до п'ятого входу блока адреси бази знань, до четвертих входів блоків координати сліду та вершини сліду, до третіх входів блоків адреси у вхідному та вихідному масиві, до третіх входів блоків ітерації, істинності та лічильника магазину та до четвертого входу блока ознак інтерпретації, четвертий вихід блока управління підключено до четвертого входу реєстра адреси оперативної пам'яті та до дев'ятого входу реєстра даних оперативної пам'яті, чий вихід підключено до входів адреси і даних оперативної пам'яті, вихід якої підключено до другого входу другого блока вводу-виводу, до першого входу блока управління та до перших входів блоків імені структури, номера слова фрейму, адреси бази знань, координати сліду, вершини сліду, адреси у вхідному масиві, адреси у вихідному масиві, ітерації, лічильника магазину, ознак інтерпретації, а другий вихід другого блока вводу-виводу підключено до других входів блоків імені структури, номера слова фрейму, адреси бази знань, координати сліду, вершини сліду, адреси у вхідному масиві, адреси у вихідному масиві, ітерації, істинності, лічильника магазину, ознак інтерпретації та до перших входів реєстра адреси і реєстра даних оперативної пам'яті, а вихід блока імені структури підключено до другого входу реєстра даних оперативної пам'яті, до третього входу блока номера слова фрейму та до четвертого входу блока адреси бази знань, вихід блока номера слова фрейму підключено до третіх входів реєстра даних оперативної пам'яті та блока адреси бази знань, вихід блока адреси бази знань підключено до третього входу блока імені структури, вихід блока координати сліду підключено до другого входу реєстра адреси оперативної пам'яті та до третього входу блока вершини сліду, чий вихід підключено до четвертого входу реєстра даних оперативної пам'яті та до третього входу блока координати сліду, вихід блока адреси у вхідному масиві підключено до п'ятого входу реєстра даних оперативної пам'яті, вихід блока адреси у вихідному масиві підключено до шостого входу реєстра даних оперативної пам'яті, вихід блока ітерації підключено до сьомого входу реєстра даних оперативної пам'яті та до третього входу буферного реєстра, чий другий вхід разом із третім входом блока ознак інтерпретації підключено до виходу блока істинності, вихід блока лічильника магазину підключено до третього входу реєстра адреси оперативної пам'яті, вихід блока ознак інтерпретації підключено до перших входів блоків істинності та буферного реєстра, та до восьмого входу реєстра даних оперативної пам'яті, вихід буферного реєстра підключено до першого входу другого блока вводу-виводу та до другого входу блока управління.

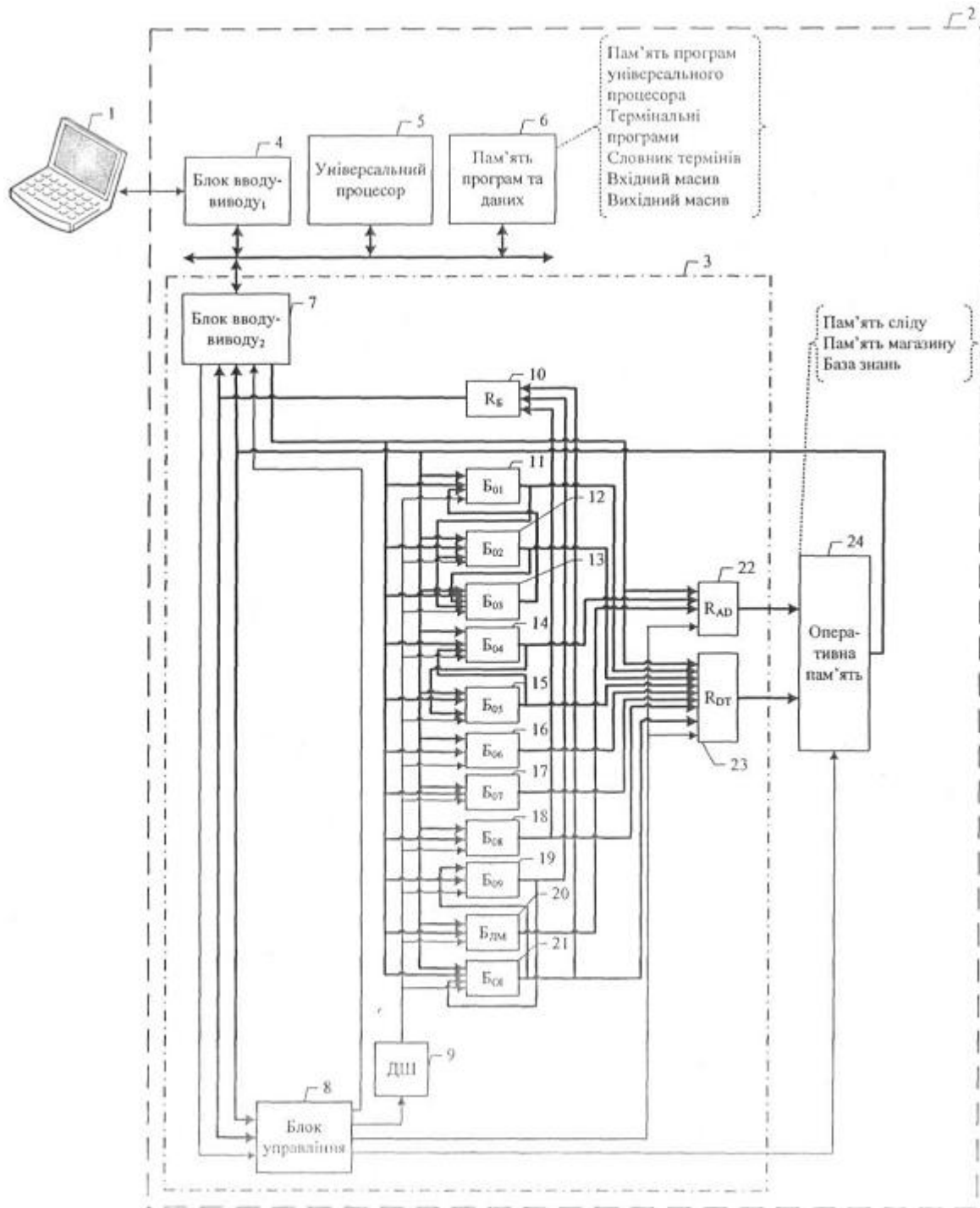


Fig. 1

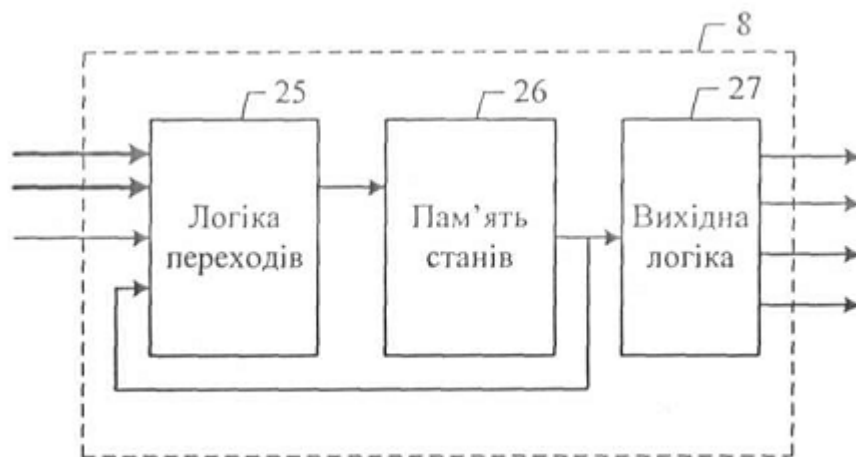


Fig. 2

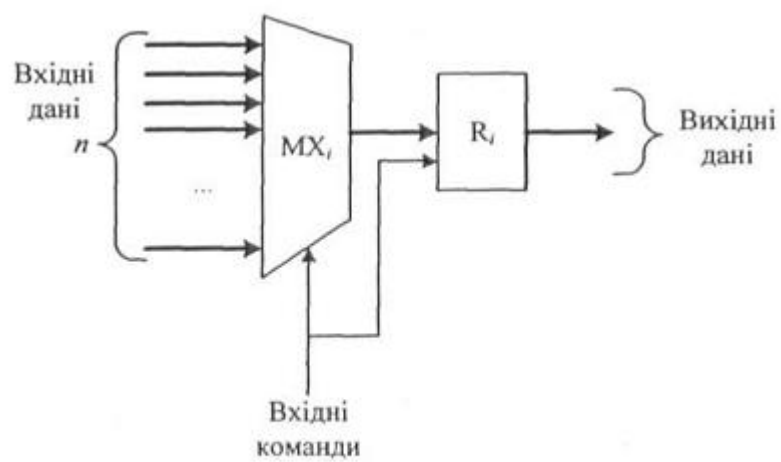


Fig. 3

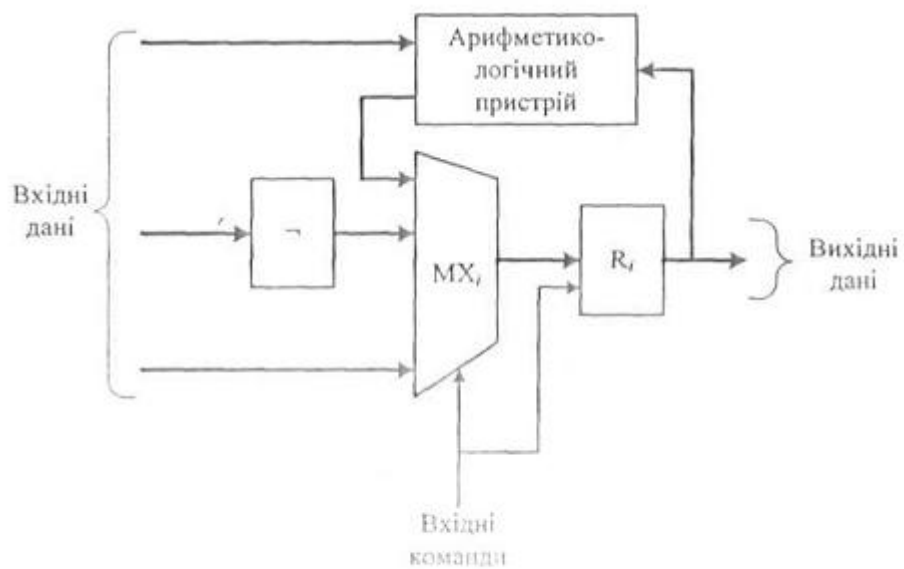


Fig. 4

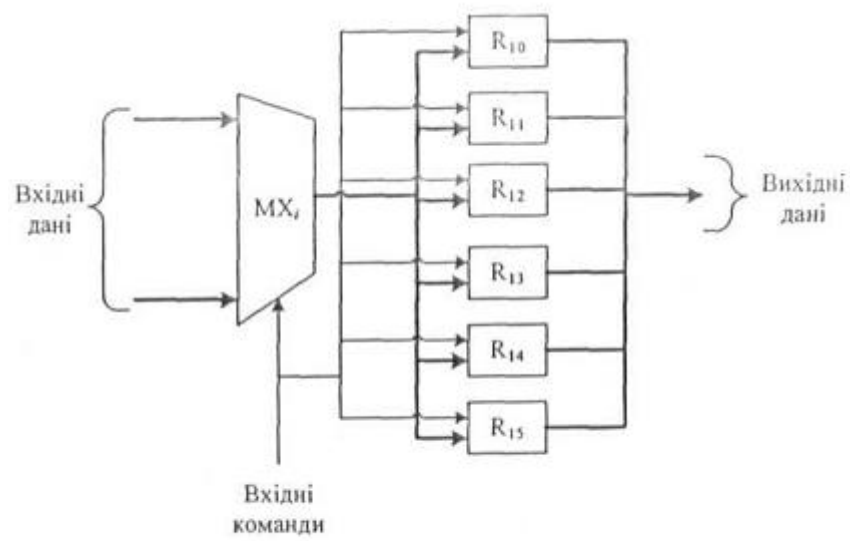


Fig. 5

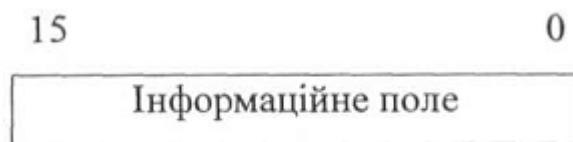


Fig. 6

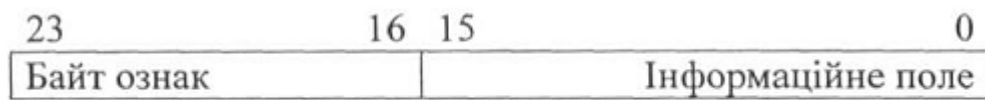


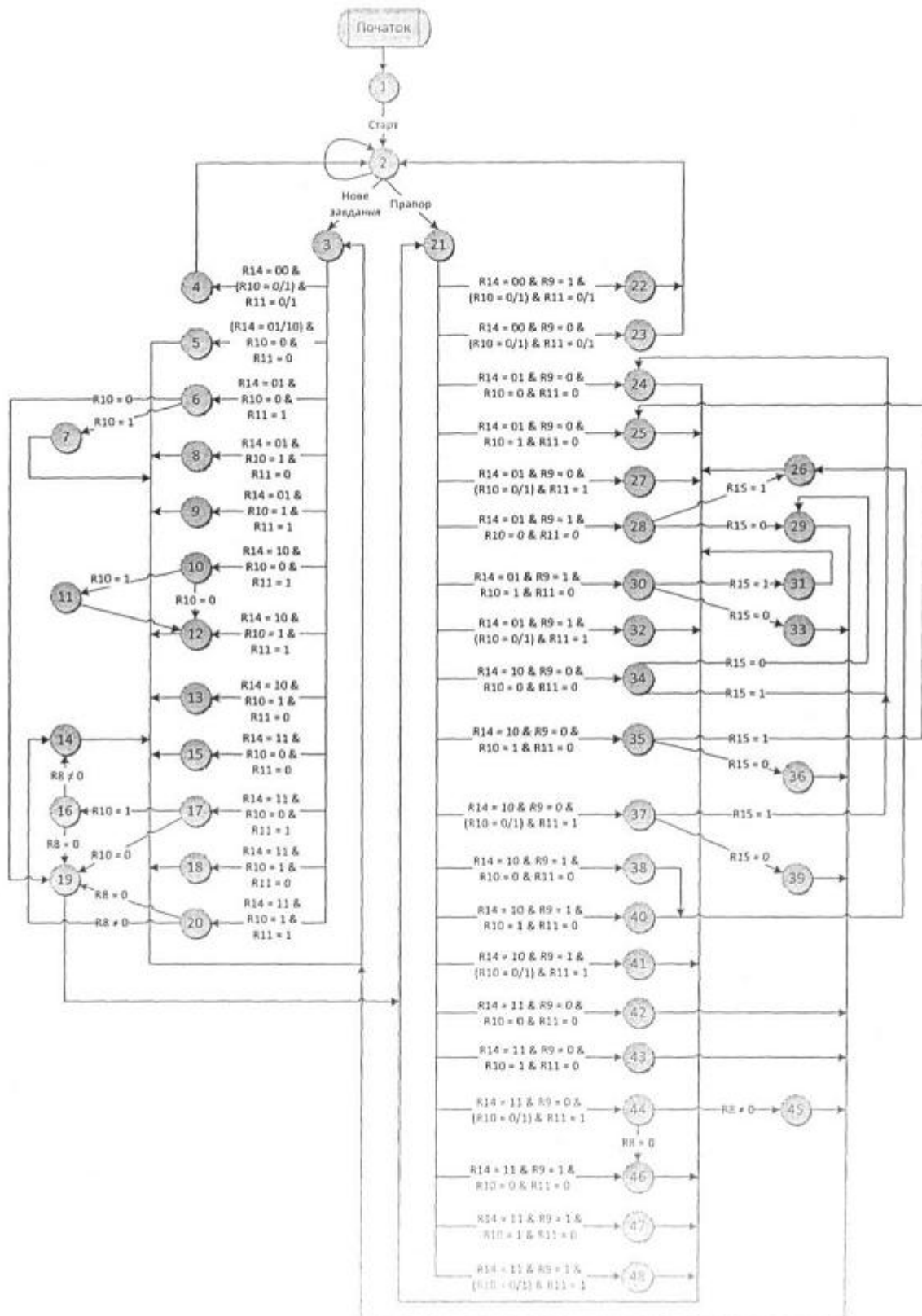
Fig. 7

Байт ознак		Опис
b_7-b_5	001	Альтернатива
	010	Послідовність
	011	Ітерація
	100	Текстова константа
	101	Термінал
b_4		Інверсія_1
b_3		Інверсія_2
b_2		Признак останнього елементу
b_1		Признак породження
b_0		Признак сліду

Fig. 8



Fig. 9



Фір. 10

Комп'ютерна верстка Д. Шеверун

Державна служба інтелектуальної власності України, вул. Василя Липківського, 45, м. Київ, МСП, 03680, Україна

ДП "Український інститут інтелектуальної власності", вул. Глазунова, 1, м. Київ – 42, 01601