

Вхідний контролер приймає і аналізує вхідний потік даних та розділяє його на лексеми. Лексеми в алгоритмах даного класу – це незалежні одиниці стиснених даних, які зазвичай можуть бути двох типів: літерали та лексична пара: зміщення/довжина. Літерал – це модифікована нестиснена одиниця даних. Пара зміщення/довжина – це ознака стисненої порції даних, яка являє собою зміщення від початку буфера та довжину повторюваного блока даних. Операційний блок відновлює лексеми від модифікації, яка могла мати місце при стисненні даних (визначається алгоритмом). Буферний пристрій зберігання лексем накопичує лексеми в тому випадку, коли кількість вихідних даних значно перевищує кількість вхідних даних, що може спостерігатися під час оброблення стиснених даних з високим коефіцієнтом стиску. Головний контролер керує роботою цілого пристрою. Він вибирає лексеми з накопичувального буфера і відправляє їх на циклічний буфер, де залежно від типу лексеми виконуються ті чи інші дії. Циклічний буфер зберігає певну кількість попередньо відновлених даних, яка використовується під час відновлення і складається з блока двоportoвої пам'яті і керуючої логіки. Керуюча логіка реалізує зацикловання буфера, приймає і обробляє лексеми.

Ця структура є гнучкою і її можна легко пристосувати для різних варіантів алгоритму відновлення. Зазвичай зміни торкаються тільки вхідного контролера та операційного блока. Крім цього, ця структура допускає багатоканальний режим роботи та переключення контексту. Для цього необхідно мати буферний пристрій для зберігання лексем та блок двоportoвої пам'яті окремо для кожного каналу.

**Висновки.** Запропонована структура системи стиску відповідає більшості вимог до систем стиску даних і може бути реалізована на сучасних ПЛІС, які містять блоки двоportoвої пам'яті. Однією з найбільших переваг цієї системи є можливість реалізувати її на сучасних ПЛІС, що спрощує тестування і скорочує терміни розроблення та введення системи в експлуатацію.

1. Lempel A. and Ziv J. *A Universal Algorithm for Sequential Data Compression* // *IEEE Transactions On Information Theory*, Vol. IT-23, №. 3, May 1977. 2. [RFC 2118] *Microsoft Point-To-Point Compression (MPPC) Protocol*. 3. [RFC 1974] *PPP Stac LZS Compression Protocol*.

УДК 681.3

А.О. Мельник, О.М. Сокіл

Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин

## **ПРОЕКТУВАННЯ ОПТИМІЗОВАНИХ СТРУКТУР КОМП'ЮТЕРНИХ ПРИСТРОЇВ ПІДБОРОМ НЕОБХІДНИХ ЕЛЕМЕНТІВ БІБЛІОТЕКИ З ВИКОРИСТАННЯМ ГЕНЕТИЧНИХ АЛГОРИТМІВ**

© Мельник А.О., Сокіл О.М., 2004

**Запропоновано новий підхід до проектування оптимізованих структур комп'ютерних пристроїв підбором необхідних елементів бібліотеки з використанням генетичних алгоритмів.**

**The new solution for an optimized structures of computer devices development, using genetic algorithms was proposed.**

**Вступ.** Проектування комп'ютерних пристроїв є досить складним процесом. По-перше, існує багато підходів до побудови структури комп'ютерного пристрою та необхідність вибору оптимальної з множини можливих структур. По-друге, є багато варіантів відображення функціональних

операторів виконуваних алгоритмів в комбінаційних схемах НВІС, тобто треба шукати компроміс між апаратними затратами та швидкістю комп'ютерних засобів. Такі особливості проектування комп'ютерних пристроїв, як складність та трудомісткість, необхідність пошуку оптимальних рішень на кожному етапі проектування, необхідність скорочення термінів проектування вимагають створення як нових методик проектування, так і систем автоматизованого проектування (САПР).

Існує багато алгоритмів пошуку оптимальної структури комп'ютерних пристроїв [1]. Самий найпростіший з них – перебір всіх можливих структур. Такий підхід є найліпшим при невеликій кількості оцінюваних структур. А коли множина оцінюваних структур сягає сотень або тисяч одиниць, то пошук перебором не реалізується на практиці. Ще одним підходом є еволюційний, який базується на еволюційній теорії [2].

Тут для розв'язання задачі пошуку оптимальної за заданими параметрами структури комп'ютерного пристрою пропонується використати генетичний алгоритм, який відображає біологічну концепцію “виживання найбільш пристосованого” [2]. Робота генетичного алгоритму починається з великої кількості потенційних рішень (в нашому випадку – це структури, сформовані з елементів бібліотеки), серед яких вибираються найкращі, які комбінуються і знову подаються на вхід алгоритму. Через певну кількість “поколінь” якість рішень підвищується і отримується оптимальна структура комп'ютерного пристрою.

**Традиційний підхід до проектування комп'ютерних пристроїв.** Проектування являє собою багаторівневий, багатокроковий та ітераційний, з поверненням назад і переглядом раніше прийнятих рішень комплексу проектних робіт, як правило, містить теоретичні та експериментальні дослідження, розрахунки та конструювання.

Функціональна декомпозиція проекту є стратегією проектування. Для системи в цілому та її блоків використовується концепція “чорного ящика”. Для “чорного ящика” розробляється функціональна специфікація, яка містить зовнішній опис блока (входи і виходи) та внутрішній опис – функцію або алгоритм:

$$F = \Phi(X, t),$$

де  $X$  – вектор вхідних величин,  $F$  – вектор вихідних величин,  $t$  – час. При декомпозиції функції  $\Phi$  розбивається на простіші функції  $\Phi_1 \dots \Phi_k$ , між якими повинні бути встановлені відповідні зв'язки, згідно з прийнятим алгоритмом реалізації функції  $\Phi$ . В результаті розбиття отримується структура.

Перехід від функції до структури є синтезом. Вибір найліпшого варіанта здійснюється згідно із результатами аналізу, коли перевіряється правильність роботи та деякі показники, які характеризують пристрій. Процедури синтезу та аналізу постійно повторюються.

Традиційні етапи проектування [3]:

- 1) системне проектування;
- 2) структурно-алгоритмічне проектування;
- 3) функціонально-логічне проектування;
- 4) конструкторсько-технологічне проектування.

На етапі системного проектування визначається архітектура, склад компонентів і основні характеристики системи, вибирається елементна база для її побудови. Це найскладніший етап, оскільки його складно формалізувати та автоматизувати.

Під час структурно-алгоритмічного проектування визначаються алгоритми функціонування апаратних і програмних компонентів системи.

На етапі функціонально-логічного проектування розробляються функціональні та принципові електричні схеми, програми, готуються тестові та контрольні дані.

На конструкторському етапі виконується прив'язка елементів проекту до конструктивних елементів.

Вище описані загальні етапи проектування комп'ютерних систем. Кожний з етапів розбивається на підетапи.

На кожному етапі проектування може бути виявлена помилковість або неоптимальність вибраного раніше варіанта реалізації або прийнятого рішення. Така ситуація призводить до

повернення і перегляду попереднього розв'язання. Розробка сучасних обчислювальних систем, які містять як мікропроцесорні рішення, так і НВІС з програмованою структурою, неможлива без застосування САПР. Враховуючи можливості сучасних САПР проектування можна рахувати закінченим після верифікації проекту в цілому, коли закінчене відлагодження готового виробу [4]. Проект або його окремі фрагменти можуть містити аналогові, аналогово-цифрові та/або цифро-аналогові елементи, будуватись на основі дискретних (цифрових) компонентів або спиратись на вбудовані мікропроцесорні засоби. Звідси випливає велика кількість варіантів проектування, які в сучасних технологіях часто називають потоком проектування (Design Flow). Наступним фактором є вибір технічної бази для реалізації фрагментів проекту, а також технічного способу реалізації самого проекту. Як правило, один і той самий пристрій може бути реалізовано різними способами.

Від підходу до проектування, на основі якого створена САПР, залежить швидкість проектування і відповідно терміни випуску продукції, тобто час від специфікації проекту до випуску перших екземплярів (Time-to-Market) [4]. Для збільшення ефективності проектування комп'ютерних пристроїв треба знайти такий підхід до проектування, який максимально спрощував би роботу проектувальникам та прискорював сам процес проектування.

**Опис підходу до проектування оптимізованих структур комп'ютерних пристроїв підбором необхідних елементів бібліотеки з використанням генетичних алгоритмів.** Як вже було сказано раніше, при великій кількості можливих реалізацій структури комп'ютерних пристроїв, процес проектування є досить громіздким та тривалим в часі. При цьому треба зробити наголошення, що потрібна ще й можливість оптимізації структури. У цьому випадку задача оптимізації структури є індивідуальною задачею оптимізації, тобто для неї задаються конкретні вхідні дані, на яких може бути отримано певний конкретний розв'язок.

Цей підхід проектування передбачає використання генетичного алгоритму (ГА), який відіграє роль як механізму оптимізації, так і механізму створення структур комп'ютерних пристроїв, та застосування бібліотеки елементів (рис. 1).

Бібліотека елементів містить множину  $N$  базових елементів, які відповідно можуть мати певну кількість (від 1 до  $n$ ) різних реалізацій. Елементи описані на одній з мов опису апаратних засобів, наприклад, VHDL. Крім того, тут міститься інформація про кожний елемент бібліотеки (його параметри), яка обробляється програмою керування бібліотекою (слідкує, щоб не виникло нестикувань між елементами, за підбором елементів відносно параметрів та ін.), коли створюється множина структур пристрою на основі цих елементів за допомогою ГА.

Як видно з рис. 1, підхід передбачає такі етапи:

- спочатку формуються вхідні дані, які описують структуру пристрою (кількість блоків, алгоритми їх функціонування, розрядності шин та ін.), вимоги щодо її проектування (наприклад, мінімізація апаратних затрат) та параметри ГА (розмір популяції, кількість батьківських пар, кількість поколінь – ітерацій та ін.);
- далі починає працювати ГА: випадковим чином формується деяка початкова множина структур пристрою (тобто автоматично генеруються відповідно до вхідних даних та за допомогою програми керування бібліотекою, яка оперує інформацією про кожний елемент, без втручання проектувальника) та оцінюється кожна з них (в бібліотеці міститься інформація про елементи, з яких вони формуються); наприклад, пристрій складається з 4 блоків, перший блок має 10 реалізацій в бібліотеці (1(1), 1(2),..., 1(10)), другий – 5, третій – 2, четвертий – 7; початкова множина (один з параметрів ГА) структур – 4 (рис. 2);
- далі виконуються послідовно операції кросовера та мутації, які описані в [2], до структур з гіршими параметрами відносно критеріїв, поставлених до структури; структура з ліпшими показниками зберігається в деякому буфері збереження оптимальних структур;
- наступним кроком є перевірка параметра ГА, який відповідає за кількість ітерацій – кількість поколінь: якщо він показує, що роботу ГА завершено, то проводиться остаточна оцінка структур в буфері збереження оптимальних структур, та вибирається структура з ліпшими показниками, які відповідають критеріям, поставленим до структури з самого

початку у вхідних даних, якщо ж ні, то робота ГА повертається до формування нової множини структур.

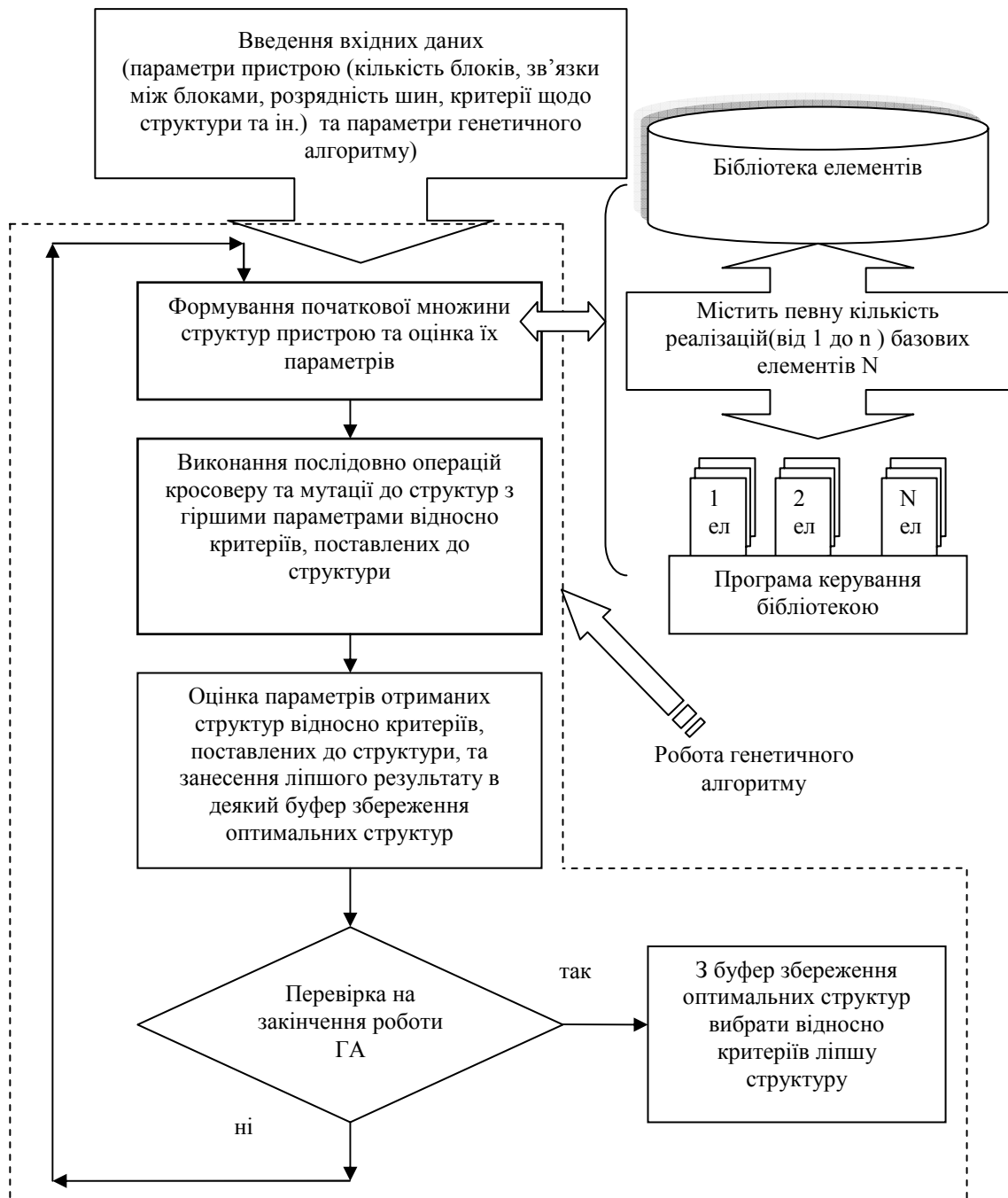


Рис. 1. Опис підходу до проектування оптимізованих структур комп'ютерних пристроїв шляхом підбору необхідних елементів бібліотеки з використанням генетичного алгоритму

Отже, таким чином можна отримати оптимізовану структуру комп'ютерного пристрою, яка відповідатиме поставленим критеріям, але не завжди є ідеальною (тобто може бути одним з варіантів реалізації структури згідно з вхідними умовами).

**Застосування підходу до проектування оптимізованих структур комп'ютерних пристроїв підбором необхідних елементів бібліотеки з використанням генетичних алгоритмів.** Для оцінки ефективності застосування вищеприписаного підходу до проектування, який використовує генетичний алгоритм, потрібно застосувати його до певних реальних проектів. Отже, була створена

модель системи проектування (принцип роботи – див. рис. 1 та 2), яка застосовується для проектування конвеєрних пристроїв. Система повинна оцінити декілька алгоритмів функціонування, щоб забезпечити побудову ефективнішої структури конвеєрного пристрою з усіх можливих варіантів, враховуючи умови, які задає проектувальник, крім того, видати характеристики всіх варіантів пристрою. Для прикладу було розглянуто проектування спеціалізованого пристрою, який реалізує функцію сортування восьми 4-розрядних чисел.

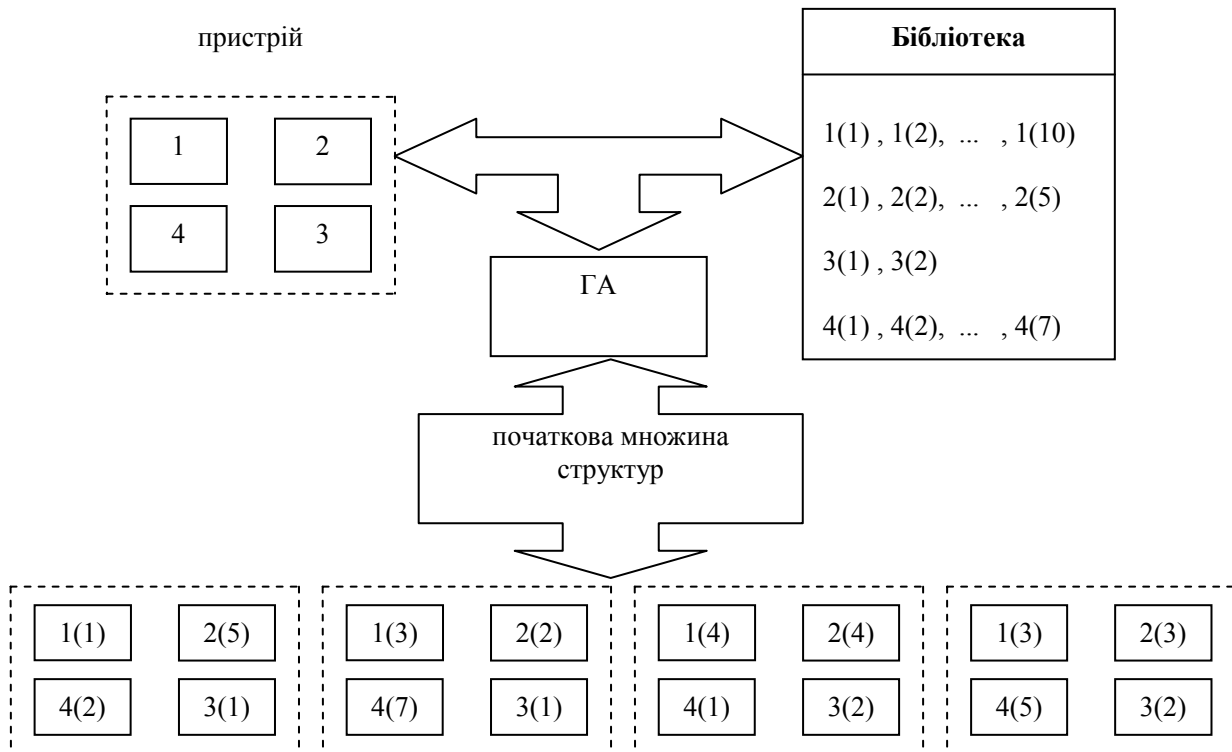


Рис. 2. Формування початкової множини структур

Постановка задачі: побудувати ефективну структури спеціалізованого пристрою сортування восьми 4-розрядних чисел, оцінити такт його роботи ( $T$ ), визначити місця розділення ярусів регістрами (вектор розміщення регістрів,  $F_{opt}$ ) для забезпечення необхідного часу виконання функції сортування при мінімальній кількості обладнання ( $W_k$ ). А також оцінити ефективність використання обладнання отриманої структури ( $E_k$ ).

Отже, отримавши таку задачу, спочатку модель системи оцінювала алгоритми, кожний з яких вже був описаний в бібліотеці. У цьому випадку алгоритмів було два. Структуру базового елемента порівняння для реалізації алгоритмів, яка була вибрана моделлю з бібліотеки елементів, показано на рис. 3.

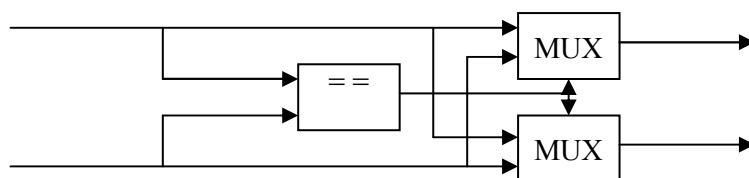


Рис. 3. Структура базового елемента порівняння, вибрана з бібліотеки елементів

Оцінюється спочатку перший алгоритм. Якщо цей алгоритм подати у вигляді потокового графу, то видно, що даний алгоритм сортування має 13 ярусів (рис. 4). Отже, найбільша довжина конвеєра, який реалізує цей алгоритм, буде рівнятися тринадцятьом. Наприклад, для апаратної

реалізації такого операційного пристрою береться програмована логічна інтегральна мікросхема Spartan XCS30XL фірми XILINX. За одиницю апаратних затрат рахується один вентиль. Таким чином можна оцінити апаратні затрати кожного ярусу конвеєра та відповідні затримки на них. Затримка на конвеєрному регістрі дорівнює 9 нс, а апаратні затрати на нього – 256 вентилів.

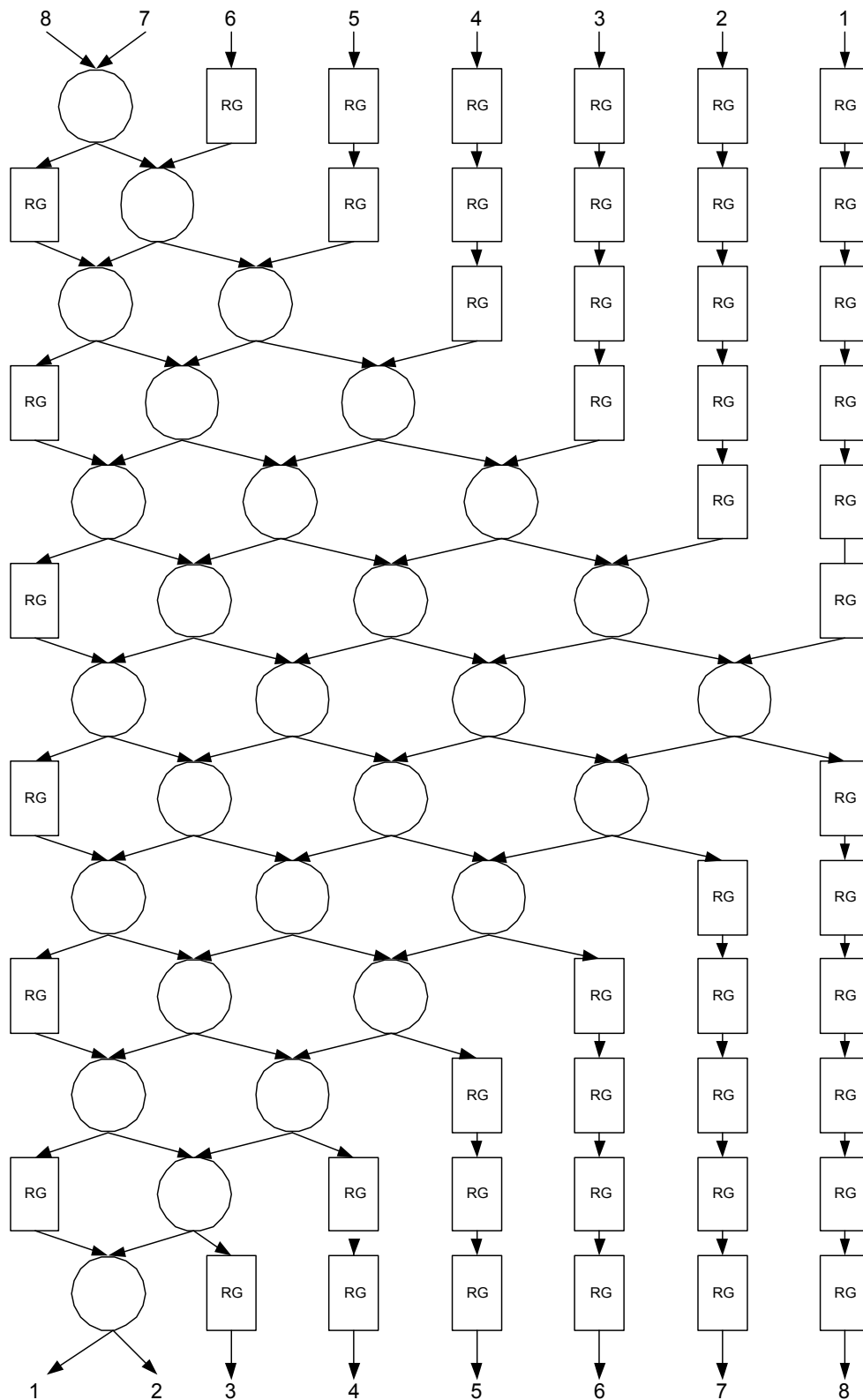


Рис. 4. Граф першого алгоритму сортування

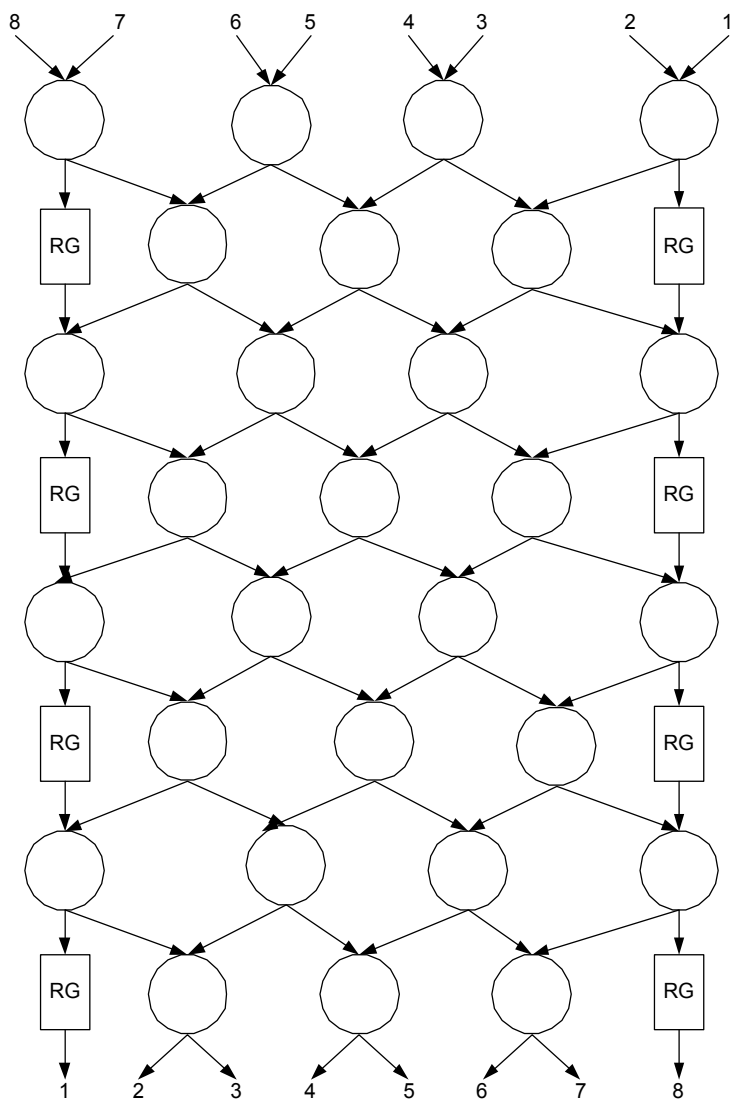


Рис. 5. Граф другого алгоритму сортування

Результати оцінки занесені в табл. 1. З результатів видно, що модель системи, основою якої є генетичний алгоритм, проектує структури досить непогано, оскільки час обробки масиву чисел близький до заданого в умові. Причому затрати на обладнання є меншими – це зумовлено обмеженням щодо кількості регістрів. Провівши зміну параметрів генетичного алгоритму, можна зробити висновок, що зі збільшенням популяції та кількості поколінь (ітерацій алгоритму або часу його виконання), результат наближається до ідеального, який в цьому випадку видав би повний перебір всіх можливих структур.

Таблиця 1

**Результати проектування першого алгоритму**

Параметри структури	Варіанти реалізації алгоритму	
	Приклади з різними вхідними даними	
	1	2
1. Вектор розміщення регістрів, $F_{opt}$	11010101011	11111111111
2. Такт конвеєра, $T_{нс}$	29	19
3. Апаратні затрати на конвеєр, $W_k$ вентилів	10820	12100
4. Ефективність використання обладнання конвеєра, $E_k$	0,1 699	0,174

Розглянемо тепер інший алгоритм сортування. Поточковий граф алгоритму зображений на рис. 5. Цей алгоритм розрахований на сортування восьми 4-розрядних чисел. Результати оцінки показано в табл. 2.

Таблиця 2

**Результати проектування другого алгоритму**

Параметри структури	Варіанти реалізації алгоритму	
	Приклади з різними вхідними даними	
	1	2
1. Вектор розміщення регістрів, $F_{\text{опт}}$	<b>01010101</b>	10110111
2. Такт конвеєра, $T$ нс.	<b>29</b>	29
3. Апаратні затрати на конвеєр, $W_k$ вентилів	<b>8580</b>	9092
4. Ефективність використання обладнання конвеєра, $E_k$	<b>0,2922</b>	0,2333

Отже, як видно при застосуванні моделі системи проектування було отримано декілька варіантів реалізації алгоритмів сортування, результатом стала структура другого алгоритму, приклад 1 (див. в табл. 2 виділене шрифтом), оскільки при його реалізації будуть менші апаратні затрати, ефективність використання обладнання є більшою.

**Висновки.** Запропонований підхід до проектування оптимізованих структур комп'ютерних пристроїв підбором необхідних елементів бібліотеки з використанням генетичних алгоритмів дає змогу значно спростити та прискорити проектування. Основними перевагами є можливість автоматичної генерації структур пристроїв, які проектуються, залежно від вимог проектувальника, та отримання нових структур за допомогою генетичного алгоритму, що дозволяє розширити простір пошуку оптимальної структури. При цьому, робота проектувальника зводиться до формування початкових та кінцевих параметрів проектованої системи (апаратні та часові параметри, алгоритм функціонування блоків системи та інші) та задання режиму проектування (налаштування генетичного алгоритму, умови вибору елементів з бібліотеки та інші). Застосування даного підходу, при проектуванні пристроїв з конвеєрною архітектурою, показало його високу ефективність, що підтверджує доцільність його застосування для проектування комп'ютерних пристроїв інших архітектур.

1. Глибовець М.М. *Основи комп'ютерних алгоритмів.* – К.: Вид. дім “КМ Академія”, 2003. – 452 с. 2. Исаев Сергей. *Популярно о генетических алгоритмах.* web: <http://www.chat.ru/~saisa/index.html>. 3. Грушвицкий Р., Мурсаев А., Угрюмов Е. *Проектирование систем на микросхемах программируемой логики.* – СПб.: БХВ-Петербург, 2002. – 608 с. 4. Норенков И.П., Маничев В.Б. *Основы теории и проектирования САПР: Учеб. для вузов по спец. “Вычислительные маш., компл., сист. и сети”.* – М.: Высш. шк., 1990. – 335.