

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»

ІВАНОВ ДМИТРО ЄВГЕНІЙОВИЧ



УДК 004.3:681.518

**ДІАГНОСТУВАННЯ ЦИФРОВИХ ПРИСТРОЇВ НА БАЗІ
ЕВОЛЮЦІЙНОГО ПІДХОДУ**

Спеціальність 05.13.05 – Комп'ютерні системи та компоненти

АВТОРЕФЕРАТ

дисертації на здобуття наукового ступеня
доктора технічних наук

Донецьк – 2013

Дисертацією є рукопис.

Робота виконана в Інституті прикладної математики і механіки Національної академії наук України.

Науковий консультант:

доктор технічних наук, професор
Скобцов Юрій Олександрович,
завідувач кафедри «Автоматизовані системи управління»,
ДВНЗ «Донецький національний технічний університет»,
Міністерство освіти і науки України.

Офіційні опоненти:

доктор технічних наук, професор
Сперанський Дмитро Васильович,
професор кафедри «Вища та прикладна математика»,
РВАТ Московського державного університету шляхів сполучень, Росія;

доктор технічних наук, професор
Кривуля Геннадій Федорович,
професор кафедри «Автоматизація проектування обчислювальної техніки»,
Харківський національний університет радіоелектроніки, Міністерство освіти і науки України;

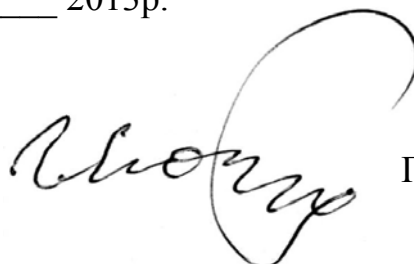
доктор технічних наук, професор
Дрозд Олександр Валентинович,
професор кафедри комп'ютерних інтелектуальних систем та мереж,
Одеський національний політехнічний університет, Міністерство освіти і науки України.

Захист відбудеться “ ___ ” _____ 2013р. о ___:00 на засіданні спеціалізованої вченої ради Д11.052.03 ДВНЗ «Донецький національний технічний університет» Міністерства освіти і науки України за адресою: 83001, м.Донецьк, вул. Артема, 58. ауд. 8.704.

З дисертацією можна ознайомитися в бібліотеці ДВНЗ «Донецький національний технічний університет» за адресою: 83001, м. Донецьк, вул. Артема, 58, II навч. корпус.

Автореферат розісланий “ ___ ” _____ 2013р.

Вчений секретар
спеціалізованої вченої ради Д 11.052.03
кандидат технічних наук, доцент



Г.В. Мокрий

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. Широке проникнення сучасної цифрової техніки в усі сфери життєдіяльності супроводжується стрімкою технологічною гонкою. З одного боку, процес виробництва НВІС, який перейшов на субмікронні рівні, дозволяє проектувати все більш складні цифрові пристрої (ЦП), включаючи, наприклад, «системи на кристалі». З іншого боку, для забезпечення надійності такої техніки потрібен відповідний розвиток методів автоматизації проектування ЦП, їх контролю, діагностики та верифікації. Одним з напрямків розвитку таких методів є еволюційні обчислення.

В даний час дослідниками розроблено ряд методів побудови діагностуючих послідовностей ЦП, що використовують у якості пошукових процедур еволюційні алгоритми (ЕА). На відміну від структурних методів, які передбачають синтез рішення, ці методи використовують техніку ймовірнісної побудови потенційних рішень та їх подальше оцінювання, зокрема на основі моделювання. Це дозволяє обробляти великі ЦП, для яких важливими також є питання енергоспоживання та розсіювання тепла.

Завдання розробки еволюційних методів контролю і діагностики розглядаються як провідними зарубіжними (Р. Prinetto, I. Pomeranz, Y. Zorian, Р. Убар, Д.В. Сперанський, К.-Т. Cheng, Е.М. Rudnick, F. Corno, D.G. Saab, L. Sekanina), так і вітчизняними авторами (В.І. Хаханов, Ю.О. Скобцов).

Однак побудову таких методів засновано на евристичному підході, вони носять точковий характер застосування та слабо взаємопов'язані між собою, що ускладнює їх ефективне застосування до вирішення нових задач. Таким чином, виникає наукова проблема об'єднання даних методів на єдиній методологічній основі, що дозволить будувати нові методи діагностування великих ЦП, які реалізуються на широкому класі сучасних паралельних обчислювальних систем (ОС).

Зв'язок роботи з науковими програмами, планами, темами. Дані дослідження безпосередньо пов'язані з виконанням: 1) Держбюджетної теми «Дослідження актуальних проблем моделювання, керування та ідентифікації дискретних систем» (1999-2003рр., №Держреєстрації 0199U001612); 2) Держбюджетної теми «Алгебраїчні, комбінаторні, логічні та еволюційні методи дослідження дискретних та неперервних систем та їх застосування до задач ідентифікації та керування» (2004-2008рр., №Держреєстрації 0104U000863); 3) Держбюджетної теми «Сучасні алгебраїчні, логічні та еволюційні методи верифікації, ідентифікації і керування дискретними і неперервними системами» (2009-2013рр., №Держреєстрації 0109U002770); 4) Держбюджетної теми «Розробка наукових основ побудови комп'ютерних систем технічної і медичної діагностики» (2007-2011рр., Н-3-07).

Мета і завдання дослідження. Метою роботи є підвищення ефективності автоматизованого діагностування великих цифрових пристроїв. Це досягається за рахунок вирішення важливої науково-практичної проблеми розробки уніфікованої методології побудови еволюційних моделей, методів та алгоритмів, в тому числі паралельних, які дозволяють вирішувати широкий клас задач діагностування та верифікації ЦП. З практичної точки зору це дозволить будувати сучасні програмні системи моделювання та діагностики цифрових пристроїв для широкого класу пара-

лельних обчислювальних систем.

Для досягнення зазначеної мети вирішуються наступні завдання:

1. Дослідження еволюційних методів з метою їх застосування в задачах побудови та оптимізації ідентифікуючих послідовностей (ІдП) ЦП.

2. Розробка моделей одно- та дворівневих ЕА побудови вхідних послідовностей діагностування ЦП, розробка на їх основі методів і алгоритмів генерації ідентифікуючих послідовностей різних класів з еволюцією одного і декількох рішень.

3. Дослідження та розробка компонент ЕА побудови та оптимізації ідентифікуючих послідовностей: кодування особин і популяцій, оцінюючих функцій, застосовуваних евристик тощо.

4. Дослідження і розробка методів оцінки розсіювання тепла для різних режимів роботи ЦП, заснованих на генетичному алгоритмі (ГА-методів); розробка ГА-методу і алгоритму побудови енергоефективних тестів, спрямованого на зменшення розсіювання тепла при тестуванні ЦП.

5. Розробка паралельних генетичних алгоритмів (ПГА) побудови ІдП на підставі схем «господар-робітник» і «острівів» з метою їх адаптації для роботи на сучасних паралельних обчислювальних системах (ПОС) різних класів, підвищення їх швидкодії та якості пошуку.

6. Розробка паралельних методів моделювання ЦП з пошкодженнями з метою адаптації ЕА для роботи на сучасних паралельних ОС з загальною та розподіленою пам'яттю.

7. Розробка автоматизованої програмної системи моделювання та діагностики, яка містить модулі еволюційного діагностування цифрових пристроїв.

Об'єктом дослідження є процес діагностування, контролю та верифікації цифрових пристроїв, заданих на структурному рівні у вигляді правильної мережі логічних елементів.

Предметом дослідження є еволюційні методи побудови і оптимізації вхідних ідентифікуючих послідовностей цифрових пристроїв.

Методи досліджень: методи та моделі технічної діагностики та перемикальних схем, методи математичного моделювання, еволюційних обчислень, теорії булевих функцій, кінцевих автоматів, теорії множин.

Наукова новизна отриманих результатів полягає в розробці єдиної методології побудови нових ефективних еволюційних методів діагностування цифрових пристроїв, що включає компонентний синтез таких методів на підставі моделей і шаблонів еволюції рішень та широке застосування паралельних обчислень.

Нові наукові результати полягають у наступному:

1. Вперше запропоновано уніфіковану методологію побудови еволюційних моделей, методів і алгоритмів, яка дозволяє на її основі розробляти нові ефективні методи генерації ідентифікуючих послідовностей, включаючи паралельні, здатні обробляти цифрові пристрої великої розмірності.

2. Отримали подальший розвиток еволюційні методи побудови ідентифікуючих послідовностей цифрових пристроїв на основі базових проблемно-орієнтованих компонент цих методів, включаючи кодування рішень та оцінюючі функції, що дає можливість підвищити їх ефективність і розробляти на їх основі

нові методи.

3. Вперше введено поняття функцій поведінки, активності і відмінності компонент цифрових пристроїв, що дає можливість при побудові оцінюючих функцій формалізувати інформацію про поведінку пристроїв на структурному рівні, більш точно оцінювати послідовності, що будуються, і, отже, поліпшити збіжність еволюційних методів.

4. Отримали подальший розвиток методи проектування енергоефективних цифрових пристроїв на основі генетичних алгоритмів, що дає можливість будувати перевіряючі тести з меншим розсіюванням тепла в процесі тестування, а також забезпечує швидке отримання точних оцінок пікового розсіювання тепла в залежності від режимів роботи пристрою.

5. Отримали подальший розвиток методи побудови паралельних генетичних алгоритмів за схемою «господар-робітник», які відрізняються організацією процедур моделювання цифрових пристроїв у функціях оцінки особин-послідовностей, що дозволяє адаптувати їх на сучасні паралельні обчислювальні системи різних класів, а також істотно підвищити швидкість роботи таких методів.

6. Отримали подальший розвиток методи побудови паралельних генетичних алгоритмів за схемою «острів» за рахунок розробки нових методів роботи компонент сервера і клієнтів, що дозволяють реалізовувати різні структури взаємодії і стратегії адаптації параметрів, проектувати методи на паралельні обчислювальні системи різної топології, а також підвищити якість і швидкість пошуку рішень.

7. Отримали подальший розвиток паралельні методи моделювання цифрових пристроїв на основі розбиття списку несправностей за рахунок їх адаптації до паралельних обчислювальних систем із загальною і розподіленою пам'яттю, що дозволяє істотно підвищити швидкодію таких методів.

8. Вперше розроблено паралельний метод моделювання цифрових пристроїв з пошкодженнями для багатоядерної обчислювальної системи з загальною пам'яттю, в якому виконується паралельне моделювання груп несправностей для кожного вхідного набору, що забезпечує істотне підвищення швидкодії такого моделювання.

9. Отримав подальший розвиток еволюційний підхід до ідентифікації цифрових пристроїв на основі застосування методу симуляції відпалу, що дає можливість розробляти нові ефективні методи побудови ідентифікуючих послідовностей.

Практичне значення одержаних результатів полягає в наступному:

1) розроблено нову систему моделювання і діагностування «АСМІД-Evolution», в якій моделі і методи побудови ідентифікуючих послідовностей доведені до алгоритмічної реалізації; дана система дозволяє будувати тестові послідовності з більш високими якісними характеристиками, що показують результати експериментів зі схемами з міжнародного каталогу ISCAS-89:

- повноту перевіряючих тестів підвищено на 7-10%, ступінь верифікації підвищено на 12-15%, параметр розсіювання тепла під час тестування знижено на 75-93%;
- швидкодію методів генерації ідентифікуючих послідовностей підвищено в 1.6-2.2 рази на двоядерній системі, в 1.1-3.2 рази на 4-ядерній системі, в 2.2-13.5 рази на 12-ядерній системі;
- швидкодію методів моделювання з пошкодженнями підвищено в 2.8-3.4 рази на 4-ядерній системі, в 4.6-6.7 рази на 12-ядерній системі, в 4.4-7.0 рази на 8-

процесорному кластері;

2) проведено машинні експерименти, на підставі яких визначено раціональні параметри розроблених евристик, що дозволяють досягти високу ефективність запропонованих методів;

3) запропонована методологія дозволяє розробляти нові еволюційні методи генерації та оптимізації ідентифікуючих послідовностей різних типів, що реалізуються на сучасних паралельних інструментальних платформах різних класів;

4) наукові та практичні результати впроваджено у ПАТ «СКТБ РТУ» при ВАТ «Топаз» і НВМП «Амплітуда», про що отримані акти впровадження; використано при виконанні Міжнародних Європейських проектів REASON - REsearch and Training Action for System On Chip DesigN (IST-2000 -30193) і EVikings II (IST-2001-37592); впроваджено в навчальний процес в курсах «Сучасні методи діагностики цифрових систем», «Технічна діагностика цифрових систем» і «Автоматизація проектування комп'ютерних систем» кафедри АСУ ДВНЗ «Донецький національний технічний університет».

Особистий внесок здобувача. Всі результати, викладені в дисертації, отримані здобувачем самостійно. У роботах, опублікованих із співавторами, здобувачу належать наступні результати: [9] - метод оцінки пікового розсіювання тепла, реалізація, експерименти; [14-17, 23] - методи симуляції відпалу побудови ідентифікуючих послідовностей різних класів; [22] - методи роботи сервера і клієнтів паралельного генетичного алгоритму, побудова та аналіз граф-моделей методів; [25] - генетичний алгоритм логічної ініціалізації; [26-28, 42-45] - розробка структур і методів роботи розподілених алгоритмів моделювання та генерації тестів, реалізація, експерименти; [29, 33] - розробка механізмів адаптації еволюційних параметрів в генетичних алгоритмах генерації тестів; [30, 46-47] - розробка генетичного алгоритму генерації функціональних тестів, реалізація, експерименти; [31-32, 36, 48] - методи генерації тестів на основі генетичного алгоритму, компоненти методів; [34] - розробка генетичних алгоритмів побудови тестів структурного та функціонального рівнів; [35] - розробка механізмів адаптації еволюційних параметрів генетичних алгоритмів; [37] - генетичний алгоритм вибору підмножини несправностей; [38] - оптимізація процедур побудови нових популяцій і моделювання в методі генерації тестів на основі генетичного алгоритму.

Апробація результатів дисертації. Основні наукові та практичні результати роботи доповідалися та обговорювалися на 45 конференціях різного рівня, основними з яких є: 8th, 9th, 10th Biennial Baltic Electronics Conference, Tallinn, Estonia, 2002, 2004, 2006; IEEE East-West Design&Test Workshop, Sochi, Russia, 2006; 10th European Test Symposium, Tallinn, Estonia, 2005; 8th International Workshop on Boolean Problems, Freiberg, Germany, 2008; 18th, 19th, 20th Conference on applied and industrial mathematics, CAIM, Iasi, Romania, 2010, 2011, Chisinau, Moldova, 2012; 1st, 2^d, 4th, 5th, 6th International scientific and technical conference «Dependable systems, services and technologies» (DESSERT), Poltava, 2006, Kirivograd, 2007, 2009, 2010, Sevastpol', 2012; «Штучні інтелектуальні системи та інтелектуальні САПР», Москва, 2002, 2003; Міжнародна конференція «Комп'ютерні науки та інформаційні технології», Саратов, Росія, 2002, 2007, 2012; 5-а, 8-а Всеросійська конференція з міжнародною участю «Нові інформаційні технології в дослідженні складних структур», Росія,

Іркутськ, 2004, Томськ, 2010; International Conference «Modern problems of Radio Engineering, Telecommunications and Computer Science», Slavsko, Ukraine, 2006; Міжнародна конференція з управління «Автоматика-2002, 2010», Донецьк, 2002, Харків, 2010; X Міжнародна науково-технічна конференція «Штучний інтелект. Інтелектуальні системи» (ШІ-2009), Дивноморське, Росія, 2009; XIX міжнародна науково-практична конференція MicroCAD-2011, Харків, 2011; Другий науково-технічний семінар «Практика та перспективи розвитку інституційного партнерства», Донецьк, 2001; Міжнародна науково-практична конференція «Розподілені комп'ютерні системи, РКС 2010», Київ, 2010; 14 міжнародна школа-семінар з перспективних систем управління, Алушта, 2001.

Публікації. За темою даної дисертаційної роботи опубліковано 80 друкованих праць, з яких 1 одноосібна монографія, 37 у виданнях відповідно до вимог МОН України (з них 17 без співавторів), 24 у збірниках праць конференцій, 26 за кордоном.

Структура та обсяг дисертації. Дисертаційна робота має 366 сторінок основного тексту, 65 рисунків, 34 таблиці. Структурно робота включає вступ, 8 розділів з висновками, основні висновки, список використаних джерел з 279 найменувань на 30 сторінках, 2 додатки на 27 сторінках.

ОСНОВНИЙ ЗМІСТ

У вступі викладені актуальність теми дисертаційного дослідження, його цілі і завдання, наукова новизна та практична цінність отриманих результатів, зв'язок з науковими темами. Представлена інформація про публікації та особистий внесок здобувача, відомості про апробацію та впровадження результатів.

У розділі 1 «Предмет дослідження та сутність наукової проблеми» охарактеризовано предмет дослідження, а також проведено аналітичний огляд сучасного стану та тенденцій розробки методів побудови ІдП різних класів та їх оптимізації.

В якості основної моделі в роботі вибрано синхронний послідовнісний ЦП, заданий на структурному рівні представлення у вигляді правильної логічної мережі. Саме використання інформації про поведінку ЦП на структурному рівні дає перевагу в порівнянні з абстрактними методами.

Аналіз структурних методів побудови ІдП показує, що вони мають неприйнятні характеристики (часові та ємнісні) при обробці великих ЦП. Це пов'язано з застосовуваними в них техніками синтезу рішень: побудова дерев обходів або обробка бульових функцій, що представляють ЦП. Дані техніки були запропоновані спочатку для комбінаційних ЦП, а потім адаптовані для послідовнісних. Їх альтернативою стали методи, які для оцінки послідовностей використовують моделювання; до них, зокрема, належать різні ЕА. На цей час цілий ряд авторів запропонували різні підходи побудови ІдП за допомогою еволюційних алгоритмів, серед яких найбільш широко представлено генетичний алгоритм. Розроблювані генетичні алгоритми побудови вхідних ідентифікуючих послідовностей використовують велику кількість евристик, що добре видно, наприклад, при побудові оцінюючих функцій. При цьому взаємопов'язувальна методика побудови таких методів та алгоритмів відсутня, незважаючи на спорідненість задач, що вирішуються.

Залежно від цілей задач та необхідної якості рішення в ЕА використовується як

справне моделювання, так і моделювання ЦП з пошкодженнями. Багаторазовий ітеративний виклик процедур моделювання для оцінки особин робить ЕА досить повільними в реалізації. Найбільш перспективним підходом, який долає зазначений недолік, є побудова паралельних версій таких ЕА. Відомі ПГА побудови ІдП орієнтовані на вузький клас доступних розробникам систем, а також є поверхнево описаними. Отже, виникає завдання розробки методології побудови паралельних версій ЕА генерації ІдП, які застосовують різні схеми розпаралелювання та паралельні процедури моделювання з несправностями, та дозволяють їх реалізацію на сучасних паралельних ОС різних класів.

Одним з найбільш перспективних напрямків у розробці сучасних ЦП є зниження їх енергоспоживання. Для цього необхідні автоматизовані засоби оцінки розсіювання тепла для структурного рівня представлення ЦП. Динамічна складова такої оцінки будується на підставі числа подій в ЦП при прикладанні заданої послідовності, тобто на підставі моделювання. Таким чином, для побудови таких оцінок перспективним є застосування ГА генерації ІдП з додатковими обмеженнями, які відповідають режиму функціонування або тестування ЦП.

На основі проведеного аналізу сформульовано цілі та завдання дослідження.

У розділі 2 «Розробка однорівневих генетичних алгоритмів побудови вхідних ідентифікуючих послідовностей ЦП» введено дві моделі застосування ЕА в задачах побудови ІдП, а також практично розроблено ряд однорівневих ГА-методів.

В *однорівневій моделі* (схемі застосування) ЕА-методів (рис.1а) єдиний цикл еволюції формує метод розв'язання задачі. До методів даного класу відносяться такі, в яких рішення знаходять за один виклик ЕА.

У тому випадку, якщо складність задачі не дозволяє методу знайти рішення за один виклик ЕА пошуку, то говорять про *дворівневу модель* (дворівневу схему, рис.1б), що припускає ітеративну двофазну реалізацію. У першій фазі відбувається пошук проміжної (локальної) цілі. Якщо таку ціль знайдено, то викликається ЕА пошуку рішення для даної локальної цілі, який формує другу фазу ітерації. Будемо називати фазу 1 верхнім рівнем, а фазу 2 - нижнім рівнем ЕА. При цьому структура фази 2 методу відповідає однорівневному ЕА побудови ІдП. У завданнях побудови вхідних ІдП рішення (послідовність) часто будується за адитивним принципом, тобто дані задачі природним чином проектуються на дворівневу схему ЕА.

В роботі розглянуто одно- та дворівневі ЕА-методи як для популяційних алгоритмів (ГА), так і для методів з еволюцією одного рішення (метод симуляції відпалу, СВ).

Далі в розділі розроблено однорівневі ГА-методи побудови ІдП.

Формально ГА задається наступним чином. Нехай Ind - множина особин (потенційних рішень); $Pop = \{pop \mid pop \subseteq Ind, |pop| < \infty\}$ - множина популяцій кінцевого розміру. Тоді ГА є упорядкована сукупність об'єктів:

$$GA = (Ind, Sel, Cross, Mut, O, Fit, pop_{нач}, N_{особ}, l, P_{скр}, P_{мут}),$$

де: $Sel: Pop \rightarrow Ind$ - операція селекції така, що для всіх $pop \in Pop$ $Sel(pop) \in pop$; $Cross: Ind \times Ind \rightarrow Ind$ - операція схрещування; $Mut: Ind \rightarrow Ind$ - операції мутації; $O: Ind \rightarrow R$ - оцінююча функція; $Fit: Ind \times Pop \rightarrow R$ - фітнес-функція; $pop_{нач} \subseteq Ind$

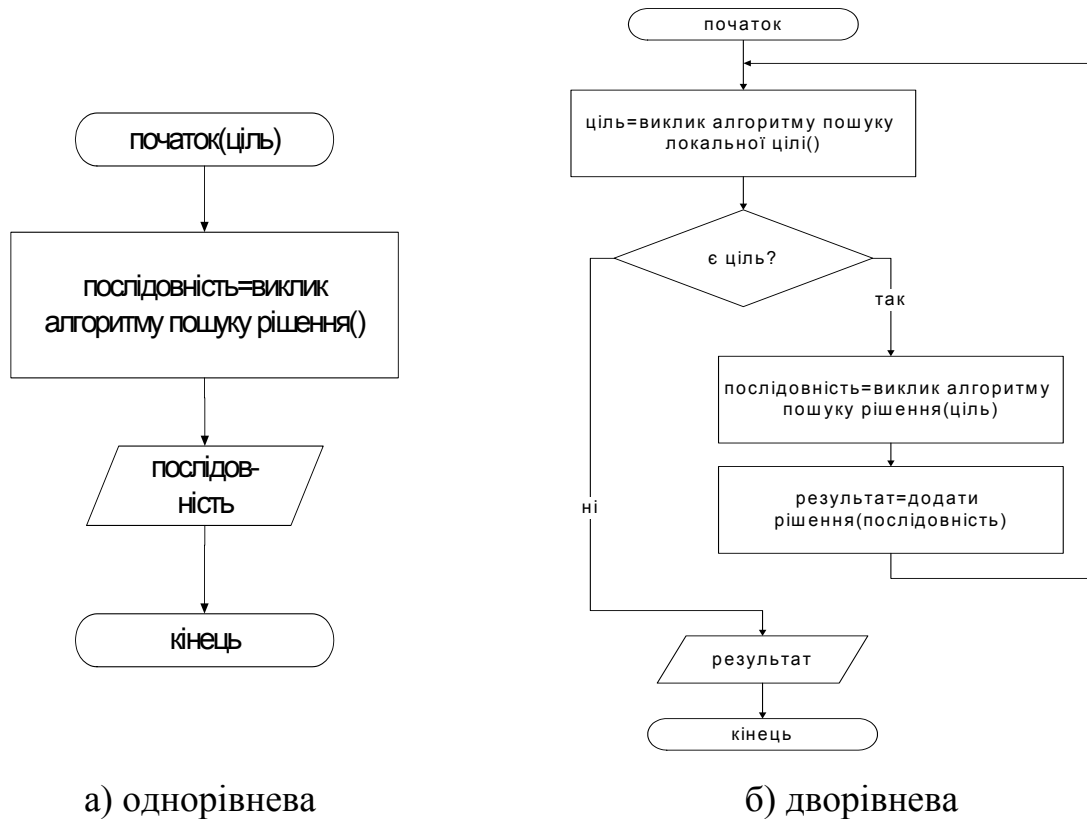


Рис.1. Моделі застосування ЕА.

- початкова популяція; $N_{особ}(pop) = |pop|$ - число особин в популяції; l - довжина особини в бітах; $p_{скр}$ і $p_{мут}$ - ймовірності застосування операцій схрещування і мутації. Оцінююча функція показує якість особини в сенсі розв'язання задачі, тоді як фітнес-функція - якість особини відносно інших в популяції. Мета ГА - пошук особини з найвищою оцінкою: $O \rightarrow \max$.

Для реалізації конкретного ГА-методу побудови ІдП ряд його компонент необхідно задавати конструктивно: еволюційні операції, оцінююча функція, спосіб побудови фітнес-функції тощо. При цьому в кінцеву реалізацію методу входить той вид операції, функції тощо, при якому виходять найкращі числові результати. Будемо називати такі компоненти *залежними від реалізації*.

Далі введено *шаблон* ГА-методів побудови ІдП, в якому залежні від реалізації компоненти повинні бути тільки названі, а їх конкретне наповнення винесено за даний шаблон. На підставі шаблону побудова нових ГА-методів генерації ІдП виконується конструктивно шляхом завдання оцінюючої функції та наповнення залежних від реалізації компонент.

В якості особини в ГА-методах побудови ІдП використовується впорядкована множина вхідних наборів. Для абстрактного рівня завдання ЦП особині відповідає вхідна послідовність $S = (X_1, \dots, X_n)$ (рис.2а), а на структурному - матриця, кожен елемент якої представляє символ з алфавіту кодування (рис.2б). Набір особин утворює популяцію.

Операції схрещування *Cross* розділимо на два класи: 1) ті, що мають аналоги

для структурного та абстрактного рівнів завдання ЦП (горизонтальне схрещування); 2) операції, які реалізуються тільки на структурному рівні і не мають явного аналога для абстрактного рівня (різні реалізації вертикального схрещування).

За аналогією операції мутації *Mut* також поділяються на два класи: додавання/видалення/зміна вектору (абстрактний та структурний рівні) та зміна стовпця (структурний рівень).

Центральним елементом ГА-методів є оцінююча функція, що формалізує мету пошуку всього алгоритму. Для її побудови використовують два типи параметрів: статичні та динамічні (табл.1).

Статичні параметри є характеристиками оброблюваного ЦП A_0 та не змінюються в процесі роботи ГА: $N_{вих}$ / $N_{вх}$ / $N_{тр}$ / $N_{к.т.}$ / $N_{ел}$ - число зовнішніх виходів / зовнішніх входів / елементів станів / контрольних точок / логічних елементів (комбінаційних блоків); $|F|$ - число несправностей, що розглядаються; H/I - керованість / спостережність елемента тощо.

Динамічні параметри є властивістю оцінюваної особини-послідовності S та обчислюються на підставі моделювання поведінки досліджуваного ЦП. У загальному випадку оцінююча функція будується з урахуванням параметрів обох типів, тобто:

$$O = O(A_0, S).$$

В задачах побудови тестів розглядається непошкоджений ЦП A_0 та множина несправних ЦП $A = \{A_1, \dots, A_n\}$, яка породжена множиною несправностей F :

$$O = O(A_0, A, S) = O(A_0, F, S).$$

Включення всіх параметрів в оцінюючу функцію при рішенні конкретної задачі може виявитися неефективним та необхідний їх відбір.

Укрупнену схему шаблону однорівневих ГА-методів побудови ІдП у вигляді

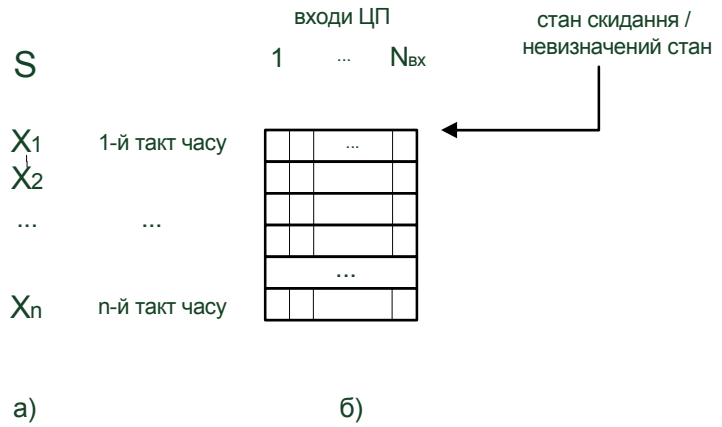


Рис.2. Кодування особини в ГА побудови ІдП.

Таблиця 1.
Динамічні параметри оцінюючих функцій.

| параметр | інтерпретація |
|---|--|
| $U_{N_{вих}}, U_{N_{вх}}, U_{N_{тр}}, U_{N_{к.т.}}, U_{N_{ел}}$ | число компонент, для яких встановлено задане значення в алфавіті моделювання |
| $R_{N_{вих}}, R_{N_{вх}}, R_{N_{тр}}, R_{N_{к.т.}}, R_{N_{ел}}$ | число відповідних компонент, для яких отримані різні значення |
| $N_{соб}$ | число подій |
| L | довжина послідовності |
| $ F_{мест} $ | число виявлених несправностей |
| $ F_{акт} $ | число активізованих несправностей |
| $N_{нер}^i$ | число множин нерозпізнаних несправностей потужності i |

псевдокоду представлено нижче.

Схема А1

```

ГА_побудови_ІдП(  $A_0$ , Параметри) {
    ПопередняОбробка(  $A_0$  );
     $Pop_{нач}$  = ПобудоваПочатковоїПопуляції(  $N_{особ}$ ,  $L$  );
    ОцінитиПопуляцію(  $Pop_{нач}$ ,  $A_0$ ,  $N_{особ}$  );
     $Pop_{тек}$  =  $Pop_{нач}$  ;
    НомерПопуляції = 0;
    while( НедосягнутоКритерійЗупинки() ) { // цикл за поколіннями
        ОбчислитиФітнесФункцію(  $Pop_{тек}$ ,  $N_{особ}$  );
        while( БудуєтьсяНоваПопуляція() ) { // построение новой популяции
            Батьки = ОпераціяСелекції(  $Pop_{тек}$  );
            Нащадки = ОпераціяСхрещування( Батьки );
            Нащадки = ОпераціяМутації( Нащадки );
            ДодаванняДоНовоїПопуляції( Нащадки );
        } // кінець while - побудова нової популяції
         $Pop_{тек}$  = ПобудуватиНовуПопуляцію();
        ОцінитиПопуляцію(  $Pop_{тек}$ ,  $A_0$ ,  $N_{особ}$  );
        НомерПопуляції++;
        АдаптаціяПараметрів();
    } // кінець while - досягнуто критерій зупинки
    СортуватиПопуляціюЗаОцінкою(  $Pop_{тек}$  );
    Рішення =  $Pop_{тек}[0]$ ; // краща особина в останній популяції
} // кінець однорівневого ГА побудови ІдП

```

Дана схема показує ітеративну побудову нових популяцій особин-послідовностей з вибраними: представленням особин, еволюційними операціями, оцінюючою функцією, параметрами еволюції. Конструктивне визначення зазначених компонент спільно зі схемою А1 означає побудову відповідного однорівневого ГА-методу.

Далі в розділі на основі даного шаблону розроблено методи логічної ініціалізації, досягнення станів та верифікації еквівалентності поведінки ЦП.

Визначення 1. Послідовність $S \in \Sigma$ називається логічно ініціалізуючою для заданого ЦП A_0 з функцією переходу δ , якщо фінальний стан $Z_i = \delta(Z_u, S) \in Z$, де Σ - множина всіх вхідних послідовностей, Z - множина всіх визначених станів, Z_u - початковий повністю невизначений стан.

Відповідно до даного визначення оцінююча функція особини S будується на підставі евристичних параметрів: n_1 - відношення числа елементів станів з визначеним значенням сигналу з $E_2 = \{0,1\}$ до загального числа елементів станів N_{mp} ; n_2 - число подій при моделюванні поведінки або активність ЦП; n_3 - довжина особини S .

Для формалізації структурної інформації про поведінку ЦП введено поняття

бульових функцій поведінки компонент. Нехай функція $g(X_j, A_0)$ показує значення на виході компоненти g в заданому алфавіті після моделювання A_0 на наборі X_j .

Визначення 2. Функція установки в задане значення $U^{(v)}(g, X_j, A_0)$ дорівнює 1, якщо після моделювання A_0 на вхідному наборі X_j вихід елемента g дорівнює v :

$$U^{(v)}(g, X_j, A_0) = \begin{cases} 1 \Leftrightarrow g(X_j, A_0) = v; \\ 0, \text{ інакше.} \end{cases} \quad (1)$$

Оскільки в якості основного в роботі обрано тризначний алфавіт E_3 , то цільовими є значення 0 і 1 (що знімають невизначеність початкового стану) та використовуються функції $U^{(0)}$ та $U^{(1)}$.

Для n_1 маємо:

$$n_1 = \left(\sum_{g \in Z} (U^{(1)}(g, X_j, A_0) \vee U^{(0)}(g, X_j, A_0)) \right) / N_{mp},$$

де Z - множина елементів станів ЦП.

Функція активності N_{cob} ЦП A_0 при моделюванні на вхідній послідовності S :

$$n_2(A_0, S) = N_{cob}(A_0, S) = \sum_{j=0}^{\text{довжина}(S)} N_{cob}(A_0, X_j) = \quad (2)$$

$$\sum_{j=0}^{\text{довжина}(S)} \sum_{g \in G} ((U^{(1)}(g, X_{j-1}, A_0) = U^{(0)}(g, X_j, A_0)) \vee (U^{(0)}(g, X_{j-1}, A_0) = U^{(1)}(g, X_j, A_0))),$$

де G - множина логічних елементів ЦП.

Параметр n_3 залежить тільки від послідовності: $n_3 = n_3(S) = \text{довжина}(S)$.

З урахуванням вагових коефіцієнтів k_1, k_2, k_3 можна записати:

$$O(A_0, S) = (k_1 * n_1(A_0, S) + k_2 * n_2(A_0, S)) * k_3^{n_3(S)}. \quad (3)$$

Реалізація методу будується на підставі наведених вище: схеми А1 одно-рівневих ГА, представлення рішень у вигляді матриці, еволюційних операцій та оцінюючою функції у вигляді (3).

Задача побудови послідовності досягнення стану (ПДС, для абстрактного рівня часто застосовується термін установча) відрізняється від попередньої тим, що тут задано початковий стан або навіть множину початкових станів Z_{start} .

Визначення 3. Для ЦП A_0 з функцією переходу δ послідовність $S \in \Sigma$ називається установчою в стан $Z_{кон}$ (ПДС для $Z_{кон}$), якщо для $\forall \alpha \in Z_{start} \subset Z$ виконуватиметься $Z_{кон} = \delta(\alpha, S)$.

Побудова ГА-методу розв'язання задачі виконується аналогічно описаному вище на підставі введеної схеми методу А1 та його компонент.

Для формалізації оцінки використовується відстань за Хеммінгом двійкового представлення $Z_{тек} = \delta(Z_{нач}, S)$ та $Z_{кон}$:

$$\alpha(A_0, S, Z_{нач}, Z_{кон}) = \sum_{g \in Z} ((U^{(0)}(g, X_j, A_0, Z_{нач}) = g_{кон}) \vee (U^{(1)}(g, X_j, A_0, Z_{нач}) = g_{кон})), \quad (4)$$

де: Z - множина елементів станів ЦП; значення компонент $g_{кон}$ визначають стан $Z_{кон}$. Оцінка (4) для покращення якості за аналогією з (3) може бути доповнена компонентами активності ЦП A_0 у вигляді (2) та довжини S .

При верифікації еквівалентності поведінки двох заданих ЦП використовуються різні визначення. Покажемо побудову ГА-методів верифікації для двох з них: тризначна незмінна заміна (ТНЗ) та тризначна еквівалентність (ТЕ).

Визначення 4. Значення сигналу v_1 називається покриваючим для значення сигналу v_2 , якщо пара (v_1, v_2) приймає тільки наступні значення: $(0,0)$, $(1,1)$, (u,u) , $(u,0)$ и $(u,1)$, де $1,0,u \in E_3$.

Якщо Z - початковий стан автомата A , на вхід якого подається послідовність S , то його вихідні реакції позначимо $A(Z, S)$.

Визначення 5. ЦП (автомат) A_1 називається ТНЗ для ЦП (автомата) A_2 (A_1 та A_2 перебувають у відношенні ТНЗ) тоді та тільки тоді, коли для довільної S вихідні реакції $A_1(Z_u, S)$ покривають вихідні реакції $A_2(Z_u, S)$.

Визначення 6. ЦП A_1 є ТЕ для ЦП A_2 тоді та тільки тоді, коли при їх старті з Z_u для довільної вхідної послідовності S пари вихідних реакцій на неї $(A_1(Z_u, S), A_2(Z_u, S))$ належать множині $\{(0,0), (1,1), (u,u)\}$.

Перевагою ТНЗ та ТЕ є те, що для великих ЦП дані властивості легше перевіряти шляхом моделювання, ніж формальним доказом. Тоді задача верифікації еквівалентності переформулюється: замість доведення еквівалентності двох заданих ЦП слід будувати послідовності, які показують їх нееквівалентність в сенсі заданих визначень.

Побудовано оцінюючі функції для ГА-методів рішення задач. Семантичне навантаження властивостей послідовностей, що будуються, полягає в тому, щоб породити найбільшу відмінність в поведінці верифікованих ЦП A_1 та A_2 :

$$O(A_1, A_2, S) = \sum_{j=1}^{\text{довжина}(S)} ((k_1 \cdot n_1(A_1, A_2, X_j) + k_2 \cdot n_2(A_1, A_2, X_j) + k_3 \cdot n_3(A_1, A_2, X_j))), \quad (5)$$

де: $k_1 - k_3$ - константи, що нормалізують; $n_1 - n_3$ - числові параметри, що показують сумарну відмінність поведінки A_1 та A_2 за множиною зовнішніх виходів, ліній станів та комбінаційних блоків відповідно.

Введено функцію відмінності поведінки компонент пристроїв A_1 та A_2 :

$$r_2(g, X_j, A_1, A_2) = g(X_j, A_1) \oplus g(X_j, A_2), \quad (6)$$

де: 2 – індекс в $r()$, що показує значність алфавіту моделювання.

Функція (6) згідно з визначеннями 5 та 6 розширена на тризначні логіки. Для визначення ТНЗ:

$$r'_3(g, X_j, A_1, A_2) = \begin{cases} 1, \text{ якщо } (g(X_j, A_1), g(X_j, A_2)) \in D_1; \\ 0, \text{ інакше;} \end{cases} \quad (7)$$

де $D_1 = \{(0,1), (1,0)\}$.

Для визначення ТЕ:

$$r_3''(g, X_j, A_1, A_2) = \begin{cases} 1, & \text{якщо } (g(X_j, A_1), g(X_j, A_2)) \in D_2; \\ 0, & \text{інакше;} \end{cases} \quad (8)$$

де $D_2 = \{(0,1), (1,0), (u,0), (u,1), (0,u), (1,u)\}$.

Застосовуючи функції відмінності у вигляді (7) та (8) для параметрів $n_1 - n_3$ в (5) будуються розрізняючі послідовності в сенсі визначень ТНЗ та ТЕ відповідно.

Реалізація методів верифікації поведінки двох заданих ЦП побудована на підставі визначених вище шаблону ГА-методу, представлення рішень та вигляду оцінки, що базується на виразі (5) з обраними функціями відмінності: (7) для ТНЗ та (8) для ТЕ відповідно.

Ефективність розроблених у розділі 2 методів вивчена в розділі 8.

У розділі 3 «Розробка дворівневих генетичних алгоритмів побудови вхідних ідентифікуючих послідовностей ЦП» розроблено дворівневі ГА-методи побудови таких послідовностей. Конструктивно вони задаються як об'єднання методів рішення задач верхнього і нижнього рівнів.

За аналогією з підходом вище розроблено шаблон верхнього рівня таких методів (рис.3). Центральним елементом методу є об'єкт *Ціль*. В задачах тестування він визначає несправність з множини, що розглядається $f_i \in F$ (для прикладу обрано множини одиночних константних несправностей - ОКН), для якої на нижньому рівні будується тестова послідовність S . В залежності від реалізованих підходів можлива побудова класів різних методів.

Спочатку розроблено метод побудови перевіряючих тестів з активізацією несправностей. Задача істотно спрощується в тому випадку, коли відмінність в поведінці A_0 та A_f (справного і несправного) поширюється на елементи станів Z , що називається активізацією несправності.

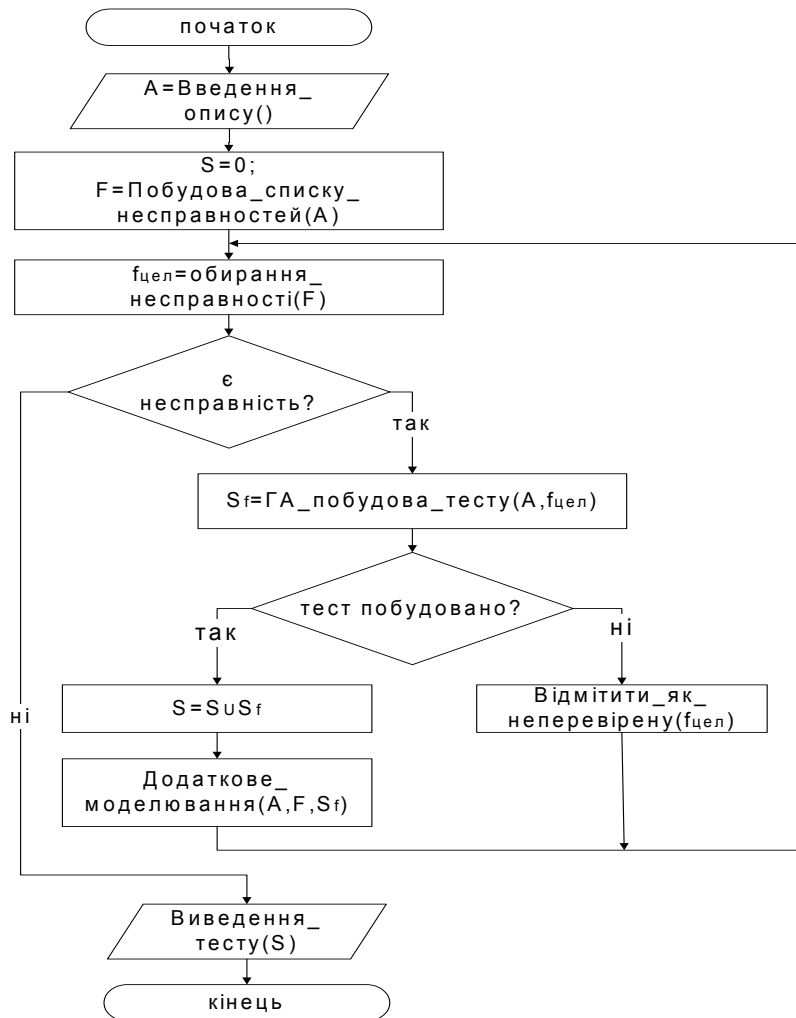


Рис.3. Шаблон верхнього рівня ГА-методів побудови тестів.

Тоді задача розбивається на дві підзадачі, які відображаються на свій рівень: активізація впливу несправності $f_i \in F$ та її вибір в якості Цілі - верхній рівень; поширення впливу f_i на зовнішні виходи ЦП - нижній рівень.

Активізація несправності f_i на верхньому рівні виконується шляхом ітеративної генерації наборів псевдовипадкових послідовностей. Активізуюча для f_i послідовність передається на нижній рівень.

Задача нижнього рівня фактично є задачею верифікації поведінки пристроїв A_0 та A_f , яка розглянута в розділі 2, та вирішується відповідним методом. При цьому в оцінку (5) можна не включати відмінність за зовнішніми виходами A_0 та A_f , оскільки вона показує рішення задачі.

Розглянутий дворівневий ГА побудови тестів не накладає ніяких обмежень на модель пошкоджень і, отже, поширюється на випадки будь-яких інших типів несправностей, для яких існують методи їх логічного моделювання: затримка поширення сигналів, перехресні несправності тощо.

Далі розроблено метод побудови тестів, в якому на нижньому рівні використовується ГА підтвердження стану. Структура верхнього рівня такого методу відповідає описаному вище за винятком того, що з множини несправностей вибирається довільна $f_i \in F$ та передається на нижній рівень.

Для побудови тесту несправності на нижньому рівні викликаються два різних методи: метод суттєвого шляху і ГА-метод досягнення стану. За допомогою методу суттєвого шляху для заданої f_i виконуються етапи просування вперед та просування назад з породженням множини альтернатив $\{(X_t, Z_t \neq Z_u)\}$, кожна з яких гарантує поширення впливу несправності на виходи ЦП. При розгляді альтернатив відбувається виклик ГА-методу досягнення станів. За його допомоги необхідно для ЦП, що стартує зі стану Z_u , побудувати послідовність, яка переводить його в стан Z_t . Відмінністю від методу розділу 2 є те, що побудована ПДС для Z_t повинна забезпечувати необхідний стан як у справному A_0 , так і в несправному A_f пристроях. Якщо ГА побудував ПДС для Z_t , то формується тестова послідовність S для f_i , що складається з наступних частин: ПДС для Z_t ; вхідний набір X_t ; послідовність поширення впливу несправності. Якщо розглянуто всі альтернативи, а тест для несправності f_i не побудований, то несправність відзначається як така, що не може бути перевірена, та не розглядається в подальшому.

Розглянутий метод використовує 3-значний алфавіт моделювання і, отже, одномірну активізацію несправності. Вочевидь, що ще більшої ефективності методу можна досягти при використанні багатозначних алфавітів та багатовимірної активізації.

Дворівнева схема може бути застосована також у випадку, коли в якості Цілі виступають більш складні структури.

Далі в розділі розроблено ГА-метод побудови діагностичних тестів.

Визначення 7. Нехай задано справний ЦП A_0 та клас несправностей $F = \{f_1, \dots, f_n\}$, який породжує клас несправних пристроїв $A = \{A_1, \dots, A_n\}$. Вхідна

послідовність, яка може відрізнити поведінку довільного пристрою A_i від поведінки A_0 , а також від поведінки всіх інших несправних пристроїв, називається діагностичною.

Ціль методу - побудова такої вхідної послідовності, яка здатна розбити вихідний клас ОКН $F = \{f_1, \dots, f_n\}$ на максимальне число класів нерозпізнаних несправностей. Позначимо через F' множину всіх класів нерозпізнаних несправностей для поточної ітерації алгоритму: $F' = \{F_1, \dots, F_j\}$. У початковий момент часу $F' = \{F\}$. Далі для кожного класу $F_j \in F'$ необхідно будувати послідовності, що розрізняють. На кінцевому етапі роботи методу послідовності будуватимуться для класів, які містять тільки по дві несправності: $|F_j| = 2$. Обраний у поточний момент часу клас $F_j \in F'$ є об'єктом *Ціль* у дворівневої моделі. Для нього на нижньому рівні за допомогою ГА будується розрізняюча послідовність. Рішенням є сукупність таких послідовностей для всіх розглянутих класів.

На верхньому рівні вибір $F_j \in F'$ проводиться за допомогою псевдовипадкової генерації послідовностей та їх оцінки. В якості *Цілі* обирається така множина F_j , для якої оцінка $O(A_0, F_j, S_i)$ (див. нижче) максимальна. Метод закінчує роботу, якщо не вдалося вибрати F_j .

На нижньому рівні побудова послідовності, що розрізняє, для F_j реалізується однорівневим ГА-методом з використанням шаблону розділу 2. Оцінка послідовності S має вигляд:

$$O(A_0, F_j, S) = \sum_{i=1}^{\text{довжини}(S)} (k_1 \cdot N_1(X_i, A_0, F_j) + k_2 \cdot N_2(X_i, A_0, F_j) + k_3 \cdot N_3(X_i, A_0, F_j)), \quad (9)$$

де $N_1 - N_3$ - параметри розрізняючої активності, які відрізняються від (5) тим, що показують розходження поведінки не двох пристроїв, а множини пристроїв, що відповідають F_j .

Введено функцію відмінності для множини порівнюваних ЦП:

$$R(g, X_i, A_0, F_j) = \begin{cases} 0, & \text{якщо виходи елементів } g \text{ множини ЦП} \\ & \text{з } F_j, \text{ однакові після подавання } X_i; \\ 1, & \text{інакше.} \end{cases} \quad (10)$$

Можна бачити, що функції відмінності $r()$ двох пристроїв (6)-(8) є окремими випадками функції відмінності $R()$ для множини ЦП (10). Функція оцінки (5) у ГА-методі верифікації еквівалентності та у методі побудови перевіряючих тестів є окремим випадком оцінки у вигляді (9). Також функція $R()$ виражається через функції досягнення значень $U^{(0)}$ та $U^{(1)}$.

Результатами роботи даного ГА-методу є: список множин F_j потужності 1 (повністю розпізнані несправності); список F' списків потужністю > 1 (кожен такий список містить нерозпізнані несправності). На підставі даної інформації

відбувається обчислення різних діагностичних мір побудованої послідовності.

Для обчислення оцінки у вигляді (9) розроблено метод діагностичного моделювання. Він будується на підставі алгоритму моделювання з пошкодженнями з наступними змінами.

- 1) Повинна підтримуватися структура даних, у якій список F' містить списки F_j нерозпізнаних несправностей.
- 2) Несправність f_l видаляється з множини F_j тільки тоді, коли вона буде відокремлена від всіх інших.
- 3) Моделювання поведінки ЦП A виконується для всіх вхідних векторів X_k послідовності S_i , $k = \overline{1, \text{довжина}(S_i)}$ та для всіх несправностей з поточного списку: $f_l \in F_j, l = \overline{1, |F_j|}$.
- 4) Після моделювання вхідного вектора X_i послідовності S_j необхідно порівняти значення на зовнішніх виходах ЦП зі значеннями виходів для всіх несправностей, які належать одному класу нерозпізнаних несправностей: $f_l \in F_j$.

Далі розроблено дворівневий метод побудови тестів для схеми вбудованого самотестування. Суть функціонального самотестування полягає у наступному. Розглянемо схему на рис.4. Центральним елементом, який виконує всі реалізовані операції з даними, є арифметико-логічний пристрій (АЛП), що є об'єктом тестування структурного рівня. Якщо розглядати АЛП A_0 як звичайний функціональний пристрій, то для K вхідних операндів X та Y ми отримуємо K результатів: $Z = X / Y$ (для прикладу вважаємо, що у поточний момент в АЛП реалізується операція ділення). Дані K результатів упаковуються в сигнатурний аналізатор та використовуються при вбудованому самотестуванні. Насправді, кожна операція у АЛП A_0 на структурному рівні виконується за деяке число машинних тактів N . У функціональному вбудованому самотестуванні передбачається, що для спостереження доступні результати роботи на кожному з тактів роботи A_0 . Таким чином, у прикладі з K вхідними операндами спостерігається $K \cdot N$ вихідних реакцій, вироблених АЛП. Отже, при

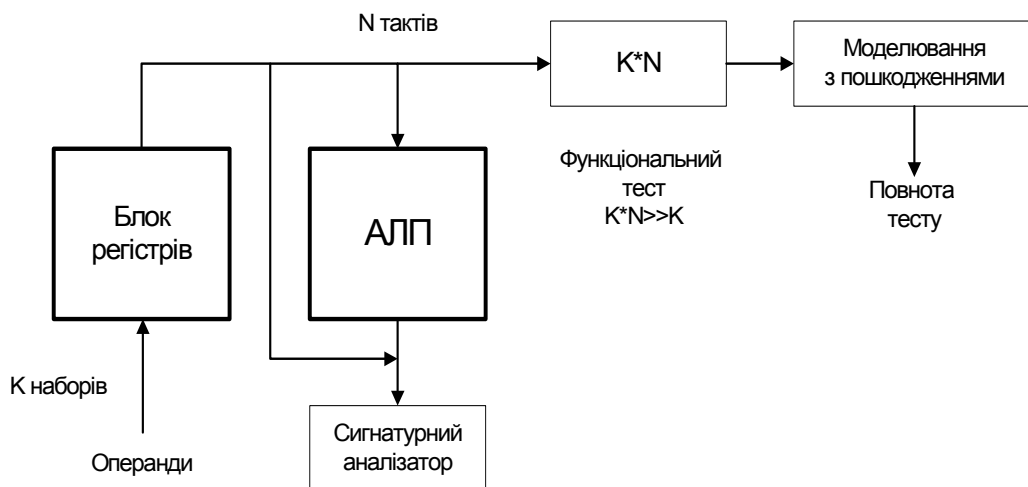


Рис.4. Схема вбудованого функціонального самотестування.

переміщенні з функціонального рівня (рівня операндів) на рівень мікроінструкцій (логічний рівень представлення ЦП) розмірність вихідних реакцій (сигнатур для аналізу) збільшується в N разів.

Задача полягає в автоматизованій побудові мінімального числа вхідних наборів, які досягають максимального покриття пошкоджень. Задача розв'язана за допомогою дворівневої моделі застосування ГА.

Довільний АЛП, що використовується на практиці, реалізує деякий базовий набір арифметичних операцій: $Op = \{op_1, \dots, op_m\}$. Зіставимо в якості об'єкта *Ціль* верхнього рівня схеми одиночну операцію. Алгоритм роботи методу верхнього рівня є простим: відбувається ітеративний вибір поточної операції $op_i \in Op, i = \overline{1, m}$ в якості *Цілі* і передача її на нижній рівень. У свою чергу для кожної обраної операції op_i з нижнього рівня повертається набір операндів, що формують тестовий набір. Підсумковий тест будується як набір тестів для всіх операцій $op_i \in Op, i = \overline{1, m}$.

Тестом для обраної операції op_i є набір операндів $S = \{(X_1, Y_1), \dots, (X_k, Y_k)\}$. Рішення задачі його побудови на нижньому рівні методу будемо будувати за допомогою однорівневого ГА. Особиною в методі служить пара операндів (X, Y) , які для пристрою A_0 представляють вхідний двійковий вектор ширини $2 \cdot l$ (l - ширина кодування операндів).

На функціональному рівні пристрій A_0 розглядається як чорний ящик, тому традиційні константні несправності не можуть бути використані в даному контексті. Обрано несправності інвертування вхідних/вихідних ліній АЛУ: будь яка інверсія в бітовому уявленні операндів X або Y повинна спостерігатися на вихідних лініях пристрою A_0 також у вигляді інверсії біта результату операції $op_i(X, Y)$. Визначимо матрицю P розмірності $2 \cdot l \times l$. У ній елемент p_{ij} дорівнює одиниці, якщо інверсія входу A_0 з номером i породжує інверсію виходу з номером j . У початковий момент тестування всі елементи матриці P дорівнюють 0. Тоді оцінююча функція набору операндів $S = \{(X_1, Y_1), \dots, (X_k, Y_k)\}$ виражається у вигляді:

$$O(op_i, S) = \sum_{i=1}^{2l} \sum_{j=1}^l p_{ij} / (2 \cdot l \cdot l) \quad (11)$$

та обчислюється після моделювання поведінки АЛП на послідовності S .

Оскільки в такій схемі операнди виступають і в ролі вхідних двійкових наборів і в ролі чисел, то застосовано два типи еволюційних операцій. Операції схрещування включають арифметичне схрещування (застосовуване з імовірністю $p_{скр}^{ар}$) та двійкове схрещування (ймовірність $p_{скр}^{бит}$), $p_{скр}^{ар} + p_{скр}^{бит} = p_{скр} < 1$. Арифметичне схрещування виконується для двох особин $A = (X_a, Y_a)$ та $B = (X_b, Y_b)$. Особиною нащадок $\tilde{A} = (\tilde{X}, \tilde{Y})$ визначається як:

$$\tilde{X} = (1 - \alpha) \cdot X_a + \alpha \cdot X_b, \quad \tilde{Y} = (1 - \alpha) \cdot Y_a + \alpha \cdot Y_b, \quad (12)$$

де $\alpha \in [0; 1]$.

Операції мутації також включають арифметичну (ймовірність $p_{мут}^{ар}$) та двійкову (ймовірність $p_{мут}^{бит}$), $p_{мут}^{ар} + p_{мут}^{бит} = p_{мут} \ll 1$. Арифметична мутація виконується над однією особиною-батьком $A = (X_a, Y_a)$ та породжує одну особину-нащадка $\tilde{A} = (\tilde{X}, \tilde{Y})$:

$$\tilde{X} = X_a \pm \Delta \cdot X_a, \tilde{Y} = Y_a \pm \Delta \cdot Y_a, \quad (13)$$

де $\Delta \in [0;1]$ - деяке невелике число.

У тому випадку, якщо АЛП реалізує цілочисельні операції, то після виконання (12) і (13) необхідно виконати округлення результату. Двійкове схрещування та двійкова мутація виконуються над двійковим поданням особин і відповідають операціям простого ГА.

Реалізація ГА-методу нижнього рівня будується на підставі шаблону методу А1 з описаними вище еволюційними операціями та оцінюючою функцією.

Оцінка ефективності методу проводилася з АЛП, що реалізує операцію ділення і включає 105 входів, 71 вихід, 513 логічних вентиля, 2383 константні несправності. Був побудований функціональний тест з семи наборів, який породжував 759 наборів структурного рівня. При цьому повнота тесту для інверсних несправностей у вигляді (11) була близько 70%, тоді як для константних несправностей структурного рівня вона склала 89.8%. Ступінь стиснення тесту при такому підході склала $759/7 > 100$.

У розділі 4 «Генетичні алгоритми в проектуванні енергоефективних ЦП» розроблено однорівневі ГА-методи оцінки розсіювання тепла для різних режимів роботи ЦП, а також підхід побудови енергоефективних тестів.

Для ЦП, що реалізується у вигляді НВІС за КМОП технологією, розсіювання тепла поділяється на статичне та динамічне. Структура проектованого ЦП впливає на динамічну складову, яка при прикладанні вхідної послідовності S з точністю до фізичних констант має вигляд:

$$A(A_0, S) = \sum_{i=1}^l \sum_g Act_g^i \cdot C_g, \quad (14)$$

де l - довжина вхідної послідовності S , C_g - число послідовників елемента g («ємність виходу»). Параметр активності логічного вентиля Act_g^i виражається через (1) та відповідає n_2 у вигляді (2).

Визначення 8 Для заданого ЦП A піковим однотактним розсіюванням тепла E_A^1 називається максимальне розсіювання тепла, яке може відбутися в ЦП за один такт модельного часу.

При обчисленні оцінки недостатньо подати на ЦП один вхідний набір, оскільки в цьому випадку початкові стани внутрішніх ліній ЦП будуть невизначеними. Тому необхідно зафіксувати деякий передпочатковий стан і подати два вхідних набори, перший з яких ініціалізує внутрішні лінії ЦП, а для другого обчислюється оцінка (14). Тоді знаходження E_A^1 зводиться до задачі максимізації:

$$E_A^1(Z_1, X_1, X_2) \rightarrow \max.$$

Рішення даної задачі виконано за допомогою однорівневого ГА. В якості особини (рис.5) використовується двійкова послідовність, яка кодує трійку об'єктів (Z_1, X_1, X_2) . В якості оцінюючої функції використовується вираз (14), причому він обчислюється тільки для другого такту роботи ЦП. Генетичні операції відповідають простому ГА.

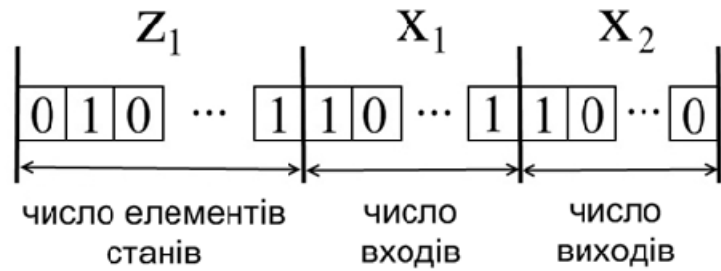


Рис.5. Кодування особини в методі оцінки пікового розсіювання тепла ЦП.

За результатами експериментів з реалізацією даного методу були побудовані більш точні оцінки E_A^1 в порівнянні з псевдовипадковою генерацією, при цьому час роботи скоротився на два порядки.

Подальшим розвитком поняття E_A^1 є пікове n -тактне та пікове стале розсіювання тепла.

Визначення 9. Для заданого ЦП A піковим n -тактним розсіюванням тепла E_A^n називається максимальне розсіювання тепла, яке може відбутися за n тактів синхронізації роботи схеми.

Визначення 10. Для заданого ЦП A піковим сталим розсіюванням тепла E_A^y називається максимальне розсіювання тепла, яке відбувається при безперервній роботі пристрою.

Вимога безперервності в останньому визначенні говорить, що послідовність внутрішніх станів пристрою A повинна замикатися через деяке число тактів модельного часу (рис.6).

Таким чином, розсіювання E_A^n залежить від $n+2$ параметрів: передпочаткового стану Z_1 та $n+1$ -го вхідного вектора і зводиться до максимізації функціоналу:

$$E_A^n(Z_1, X_1, \dots, X_{n+1}) \rightarrow \max. \quad (15)$$

У випадку E_A^y додається умова циклічності станів:

$$Z_2 = Y(X_n, Z_n). \quad (16)$$

Рішення задач оцінювання параметрів E_A^n та E_A^y також побудовано на підставі

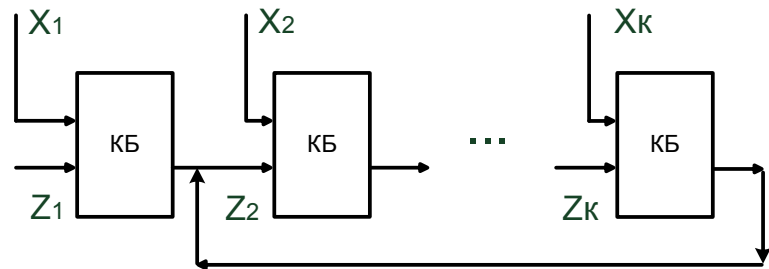


Рис.6. Розгортання КБ ЦП для оцінювання сталого розсіювання тепла.

ГА. Кодування особини (рис.5) розширено на необхідну кількість вхідних наборів. В якості оцінки для E_A^n використовується (15), а для E_A^y до нього додається обмеження (16). Реалізація запропонованих ГА-методів дозволяє будувати точні оцінки та залежності параметрів E_A^1 , E_A^n та E_A^y .

Далі в розділі розробляється підхід побудови енергоефективних тестів. Енергоефективним називається тест, при прикладанні якого до заданого ЦП розсіювання тепла буде мінімальним при заданій повноті. Задача полягає у вирішенні наступного протиріччя: з одного боку, тестування породжує найбільшу вентиляну активність в ЦП - (2), з іншого боку, це ж веде до збільшення розсіювання тепла - (14). Рішення завдання ґрунтується на надлишковому тестуванні та складається з трьох етапів.

Етап 1 підходу полягає в тому, що при заданому параметрі надлишковості тестування r необхідно побудувати множину тестових послідовностей $S = \{S_1, \dots, S_n\}$, яка: 1) має максимальну надлишкову повноту $P_r(S)$; 2) забезпечує для кожної несправності в списку перевірку не менше ніж r раз.

Рішення задачі даного етапу побудовано на підставі дворівневого ГА-методу побудови тестів (розділ 3) шляхом введення лічильника кількості перевірок для кожної несправності. Рішенням є набір послідовностей $S = \{S_1, \dots, S_n\}$.

На *етапі 2* необхідно для кожної підпослідовності $S_i \in S$ обчислити параметр розсіювання тепла $E(S_i)$ та побудувати список несправностей, які перевіряються саме даною підпослідовністю. Задача розв'язана шляхом ітеративного виклику процедури оцінки для кожної $S_i \in S$.

На *етапі 3* необхідно з множини послідовностей S вибрати таку (суб)оптимальну множину $S' \subset S$, щоб: 1) повнота тесту у традиційному (ненадлишковому) розумінні для даної множини не погіршилася:

$$P(S') = P_r(S); \quad (17)$$

2) розсіювання тепла для S' було, за можливістю, мінімальним:

$$E(S') \rightarrow \min. \quad (18)$$

Задача розв'язана за допомогою однорівневого ГА-методу. В якості особини виступає підмножина $S' \subset S$, яка задається номерами послідовностей. Операція схрещування полягає в породженні нової множини, що містить випадкові номери з двох батьків. Мутація включає видалення, додавання та заміну номерів послідовностей. Некоректні операції заборонені. Еволюція популяції поділяється на дві фази. У *першій фазі* з множини S обираються підмножини S'_i , для яких виконується умова (17), а функція оцінки має наступний вигляд:

$$f(S'_i) = P_r(S) - P(S'_i).$$

У *другій фазі* розвиток популяції йде таким чином, щоб зменшити розсіювання тепла для всіх особин-підмножин, оцінка має вигляд:

$$f(S'_i) = E(S) - E(S'_i).$$

Для вимірювання ефективності методу зменшення розсіювання тепла загалом фіксуються два параметри:

1) $E_{1 \rightarrow 2} = \min_{S_i'} E(S')$ на вході у фазу 2 алгоритму, тобто у той момент часу, коли для

всіх особин виконано умову (17);

2) $E_{кон} = \min_{S_i'} E(S_i')$ - оцінка розсіювання тепла для кращої особини кінцевої

популяції. Тоді ефективність методу обирання субоптимальної підмножини буде визначатися їх відношенням:

$$K = E_{кон} / E_{1 \rightarrow 2} \cdot 100\%. \quad (19)$$

Реалізація методу при $r = 5$ дозволила зменшити розсіювання тепла у сенсі (19) в середньому на 85.6%, при цьому час роботи ГА-методу вибору підмножини (етап 3) навіть у найгіршому випадку не перевищив однієї хвилини.

У розділі 5 «Розробка паралельних версій генетичних алгоритмів побудови вхідних ідентифікуючих послідовностей» розроблено паралельні версії ГА побудови ІдП за схемами «господар-робітник» та «островів». У ГА-методах побудови вхідних ІдП основним моментом є те, що для оцінки якості рішень, що генеруються, виконується моделювання (справне або з пошкодженнями) роботи ЦП на заданій послідовності-особині, що робить такі методи досить повільними. Для подолання цього недоліку розробляються ПГА.

Виділено наступні підходи адаптації ГА для роботи на паралельних обчислювальних системах.

1. Побудова паралельних версій ГА без зміни структури алгоритму, але з організацією паралельного обчислення в процедурі оцінки популяції: схема «господар-робітник».

2. Побудова паралельних версій ГА зі зміною його глобальної структури: схема «островів».

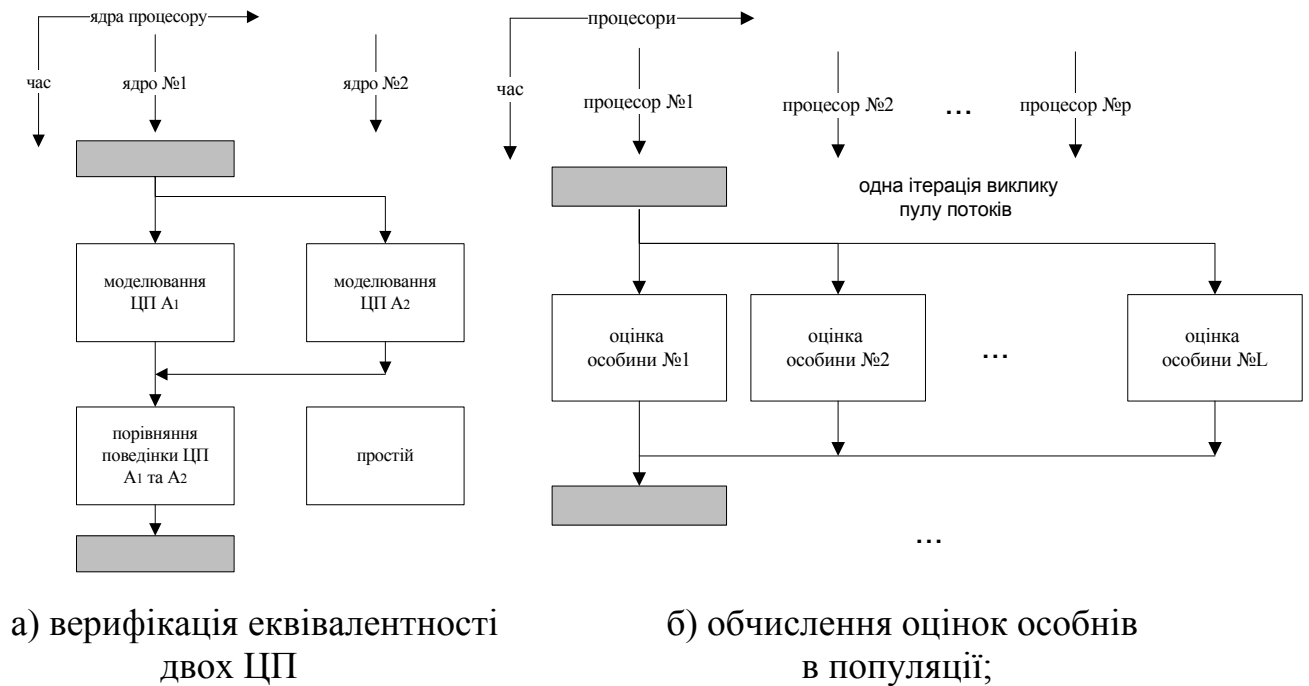
3. Побудова паралельних процедур моделювання всередині ЕА, при цьому структура алгоритму також не змінюється.

Методи, що розробляються, повинні проєціюватися на доступні розробникам ПОС, виходячи з чого виділено три основні класи таких систем:

- слабопаралельні із загальною пам'яттю: робочі станції з одним центральним процесором, що містить 2-4 обчислювальних ядра;
- сильнопаралельні системи із загальною пам'яттю: робочі станції з декількома центральними процесорами і загальним числом обчислювальних ядер 8-12;
- обчислювальний кластер: незалежні робочі станції, кожна з яких містить свою локальну оперативну пам'ять, пов'язані високошвидкісними лініями зв'язку.

Схема «господар-робітник» припускає, що в паралельній версії алгоритму існує тільки один основний цикл ГА побудови популяції, який відповідає процесору «господарю», а інші процесори називаються «робітниками» та реалізують процедури обчислення оцінюючих функцій особин. У даній схемі виділено три підходи побудови ПГА для багатоядерних ОС із загальною пам'яттю. У всіх методах в якості інструментарію при реалізації використовується багатопоточне програмування.

Перший підхід (рис.7а) полягає в паралельному моделюванні поведінки декількох ЦП всередині процедури оцінки однієї особини. Така ситуація, зокрема, виникає в методах, де для оцінки заданої послідовності-особини необхідно виконати



22.

Рис.7. Діаграма паралельного виконання потоків моделювання.

моделювання поведінки двох ЦП A_1 та A_2 . У послідовній версії функції оцінки дані ЦП моделюються одне за одним, не забезпечуючи ядра процесора навантаженням. В паралельній версії організуються два потоки, кожний з яких виконує моделювання одного ЦП. Дана схема дає найкраще прискорення для двоядерної ОС. Для великих схем каталогу ISCAS-89 експерименти з реалізацією методу показують наступні усереднені характеристики паралелізації: прискорення роботи $S_2 = 1.88$; ефективне завантаження ядер $E_2 = 0.94$; частка послідовного коду $f = 0.06$. Дані характеристики є високими, хоча і досягаються на вузькому класі ОС.

Другий підхід є більш загальним та полягає в паралельному обчисленні оцінок декількох особин в одній популяції. Діаграма виконання однієї ітерації паралельних потоків показана на рис.7б. Найкраща ефективність методу досягнута, коли число потоків не збігається з числом особин в популяції і числом обчислювальних ядер (для відображення числа потоків введено додатковий верхній індекс). Для чотириядерної ОС із загальною пам'яттю усереднені характеристики паралелізації склали: $S_4^{100} = 2.04$, $E_4^{100} = 0.51$, $f_4^{100} = 0.51$. При збільшенні числа ядер з 2 до 4 характеристики покращилися лише незначно, при цьому $i = 100$, що істотно більше числа ядер процесора.

Третій підхід є комбінацією перших двох і полягає в тому, що при паралельному обчисленні оцінок особин в одній популяції всередині кожної процедури оцінки особини також є паралельні процедури моделювання ЦП. Для діаграми на рис.7б всередині кожної процедури обчислення оцінки буде виконуватися два обчислювальних потоки. Характеристики паралелізації в цьому випадку дуже близькі до другого підходу, але з ростом числа паралельних потоків падають сильніше.

Далі даний підхід розширено на сильнопаралельні ОС із загальною пам'яттю.

Експерименти проводилися на 12-ядерній системі (два процесори Intel (R) Xeon CPU X5650 по 6 обчислювальних ядер). У серії експериментів число потоків змінювалося від 1 до 128. Графік зростання прискорення в порівнянні з однопоточною реалізацією в кращому, найгіршому та середньому випадках наведено на рис.8. Середнє за набором схем прискорення склало $S_{12} = 8.09$, максимальне $S_{12} = 13.05$. При цьому число потоків i , при якому досягається максимальне прискорення, сильно варіюється за схемами; в середньому за набором схем $i = 80$, що істотно вище числа ядер в інструментальній ОС.

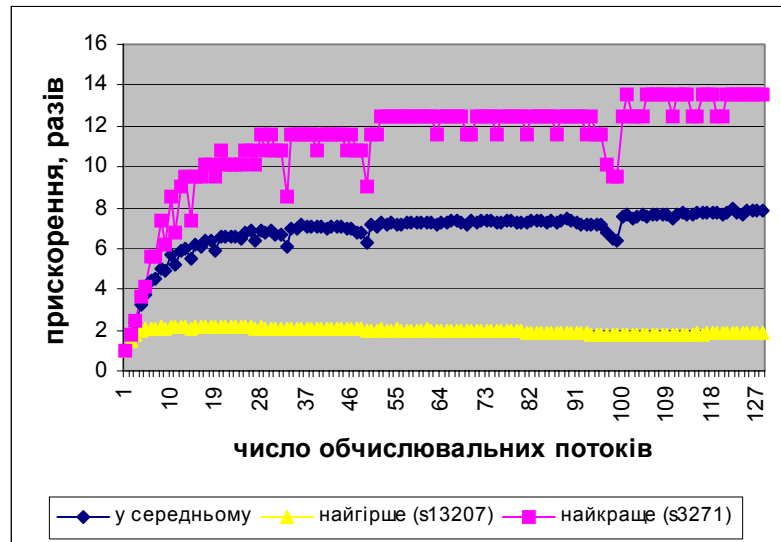
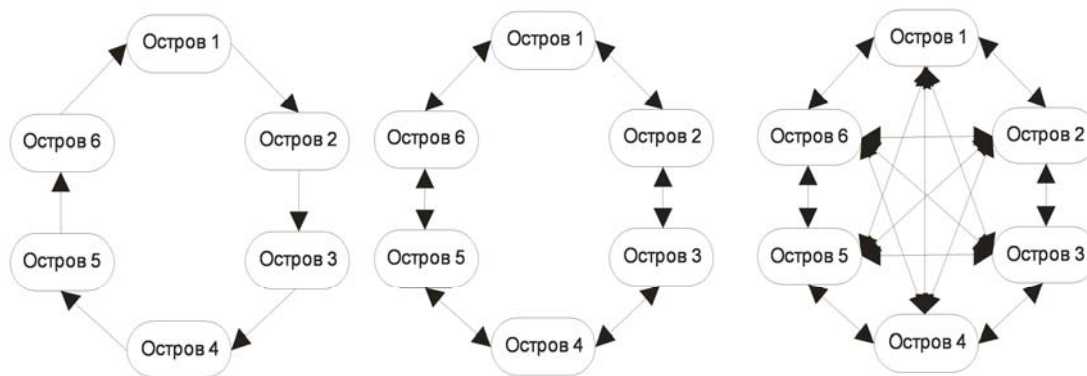


Рис.8. Графіки прискорення ПГА в залежності від числа обчислювальних потоків.

Далі в розділі розроблено схему «островів» ПГА, в якій виділено компоненти сервера і островів. Керуючі функції виконує головний процесор-сервер: побудова початкової популяції, ініціалізація ГА островів, визначення параметрів ГА островів, завершення їх роботи тощо. Тут реалізується верхній рівень ГА-методу побудови тесту (розділ 3). Острови являють собою підпопуляції рішень, на кожному з яких виконується своя копія ГА-методу побудови ІдП - нижній рівень. Генетичні параметри роботи цих ГА, зазвичай, дещо відрізняються один від одного, що дозволяє направляти пошук у відмінному від інших островів напрямку.

Суттєвим елементом такої схеми є операція обміну між островами, звана міграцією. Вона дозволяє коректувати напрямок пошуку для менш вдаливих островів. Основними її характеристиками є: топологія (рис.9), що визначає, які підпопуляції



а) по кільцю;

б) двонаправлений по колу;

в) кожний кожним;

Рис.9. Топології обміну в ПГА схеми «островів».

будуть вважатися сусідніми; час ізоляції, який визначає, скільки поколінь в ГА не буде проводитися міграція; ступінь міграції, яка показує кількість особин, що беруть участь в міграції; стратегія відбору особин для міграції; стратегія видалення особин з підпопуляції при проведенні міграції тощо.

Для компонент островів і сервера розроблено методи роботи, що дозволяють на їх основі реалізовувати довільні топології безступікової взаємодії компонент та схеми адаптації параметрів ГА по островах, а також проектувати такі методи на паралельні ОС різних класів.

У розділі 6 «Розробка паралельних методів моделювання ЦП з пошкодженнями» розроблено такі методи для паралельних ОС різної структури. В якості базової обрана схема паралелізації з розбиванням списку пошкоджень, що показує хорошу масштабованість із зростанням числа вузлів: повний список несправностей $F = \{f_1, \dots, f_n\}$, що підлягають моделюванню, розбивається на L підсписків F_1, \dots, F_L , де L - число доступних вузлів ОС; далі кожен підсписок моделюється на своєму вузлі. Моделювання на кожному вузлі ОС, в свою чергу, реалізується за допомогою паралельного за розрядами машинного слова методу моделювання з одиначним поширенням несправностей.

Спочатку розроблено метод розподіленого моделювання для ОС з розділеною пам'яттю (обчислювальних кластерів), в якому виділені два компоненти: сервер і клієнт. Сервер виконує організаційні функції: введення опису ЦП A_0 , побудова списку F несправностей, що моделюються, введення тестової послідовності S , пошук клієнтів, розбиття списку несправностей, передача даних клієнтам, очікування результатів та формування звіту. В свою чергу кожен клієнт очікує необхідні для моделювання дані, виконує безпосередньо моделювання та передає на сервер його результати. Схема взаємодії сервера і клієнтів показана на рис.10.

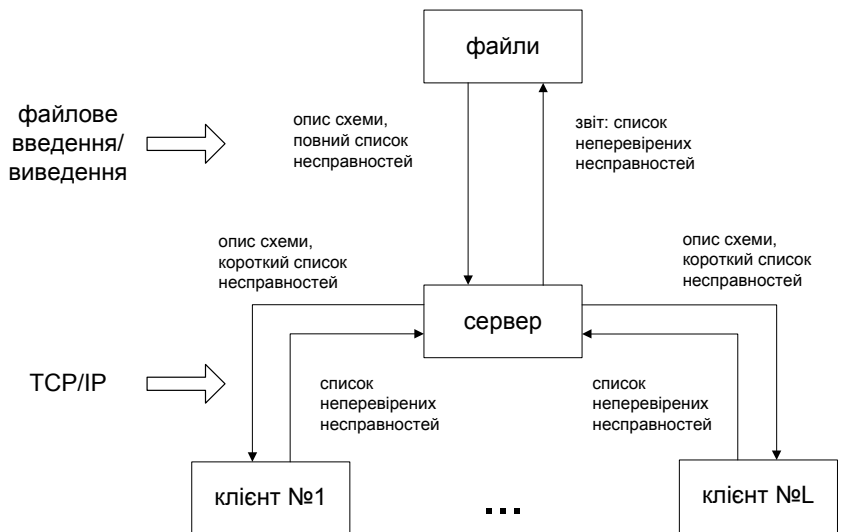


Рис.10. Взаємодія компонент розподіленого методу моделювання ЦП з несправностями.

Для даної схеми розроблено методи, що реалізують роботу сервера і клієнтів, та виконано програмну реалізацію. Апробація методу проводилася на кластерній системі з вузлами на базі процесору Intel Celeron 2400Мгц, обсяг ОЗП 256Мбайт, число вузлів =8, зв'язок за технологією 100BaseT. Для великих схем каталогу ISCAS-89 результати паралелізації наведені в табл.2. Тут: $H_i/N_i/E_i/f_i$ - прискорення / число подій моделювання / ефективність завантаження ядер / частка послідов-

Таблиця 2.

Характеристики паралелізації методу моделювання ЦП з пошкодженнями.

| Ім'я схеми | H_8 , разів | $\frac{N_8^{испр}}{N_1^{испр}}$, раз | $\frac{N_8^{неиспр}}{N_1^{неиспр}}$, раз | $\frac{N_8}{N_1}$, раз | E_8 | f_8 |
|------------|---------------|---------------------------------------|---|-------------------------|-------|-------|
| s9234 | 4.38 | 8 | 1.01 | 1.02 | 0.55 | 0.12 |
| s13207 | 5.34 | 8 | 1.01 | 1.02 | 0.67 | 0.07 |
| s15850 | 5.23 | 8 | 1.01 | 1.02 | 0.65 | 0.08 |
| s35932 | 5.14 | 8 | 1.02 | 1.04 | 0.64 | 0.08 |
| s38417 | 6.98 | 8 | 1.00 | 1.01 | 0.87 | 0.02 |

ного коду при реалізації з i обчислювальними потоками відповідно. Видно, що із зростанням розмірності схеми характеристики паралелізації поліпшуються. Головними факторами, що обмежують зростання швидкодії в даній реалізації, є час пересилки даних та число подій моделювання непошкодженого ЦП, яке росте пропорційно числу клієнтів. Це пов'язано з необхідністю на кожному вузлі відновлювати поведінку справного ЦП для подальшого порівняння з несправними.

Далі даний метод був адаптований для слабопаралельних ОС із загальною пам'яттю: процедури обміну інформацією між сервером та клієнтом були видалені, оскільки всі дані зберігаються в загальній пам'яті. Реалізація методу на 4-ядерній робочій станції дозволила підвищити швидкість моделювання в 2,81-3,44 рази в порівнянні з однопотоковою реалізацією. При цьому число потоків моделювання з несправностями має збігатися з числом обчислювальних ядер (рис.11).

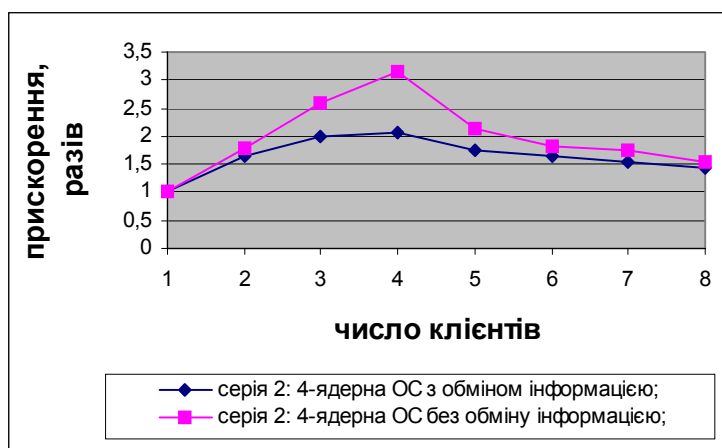


Рис.11. Прискорення роботи паралельного методу в залежності від числа клієнтів.

При запуску даних методів на сильнопаралельній ОС із загальною пам'яттю (12 обчислювальних ядер) характеристики паралелізації практично не зростали при завантаженні ядер 100%. Для систем даного класу запропоновано принципово новий метод моделювання ЦП з несправностями. Його суть полягає в тому, що для кожного вхідного набору виконується паралельне моделювання груп несправностей. У таких групах несправності моделюються паралельно за розрядами машинного слова. Діаграма обчислювальних потоків для одного вхідного набору показана на рис.12. Тут не можна паралельно виконувати моделювання справного ЦП для такту часу t , оскільки його результати вже необхідні для перевірки впливу несправностей в групах.

Далі досліджено характеристики паралелізації методу в залежності від кількості

паралельних потоків моделювання груп несправностей. На рис.13 наведено графік росту прискорення методу в кращому, середньому і найгіршому випадках щодо однопоточної реалізації. Пікові значення досягаються при числі потоків моделювання груп несправностей =11, тоді загальне число потоків дорівнює числу ядер (=12). Якщо ж організувати 12 потоків моделювання груп несправностей, а моделювання справного ЦУ виконувати окремо, то характеристики паралелізації впадуть на 7-9%.

Реалізація методу на 12-ядерній ОС дозволила досягти прискорення роботи методу в 4.61-6.72 за набором контрольних схем у порівнянні з однопоточною реалізацією.

Також в розділі одержано аналітичні оцінки масштабованості методів моделювання ЦП з несправностями. У припущенні, що число виявлених несправностей в одиницю модельного часу зменшується експоненціально, частина неперевірених несправностей після n -го набору складе:

$$U(n) = N_F (1 - \alpha r (1 - e^{-\lambda(n-1)})), \quad (20)$$

де N_F - загальна кількість модельованих несправностей, α , r , λ - евристичні константи. На підставі (20) побудовані оцінки часу моделювання для різних методів. Для однопроцесорної реалізації без розбиття списку маємо:

$$T_1(L, N_F, N_{эл}) = \gamma \beta N_{эл} L + \gamma \delta N_{эл} F L (1 - \alpha r),$$

де: γ - коефіцієнт вартості моделювання одного логічного елемента в одиницю часу, β - частина логічних елементів, необхідна для моделювання поведінки справного ЦП, δ - частина загального числа логічних елементів, які моделюються для однієї несправності, L - довжина послідовності. Видно, що час моделювання на однопроцесорній ОС виражено сумою двох доданків, де перший показує час моделювання справного ЦП, а другий - несправних.

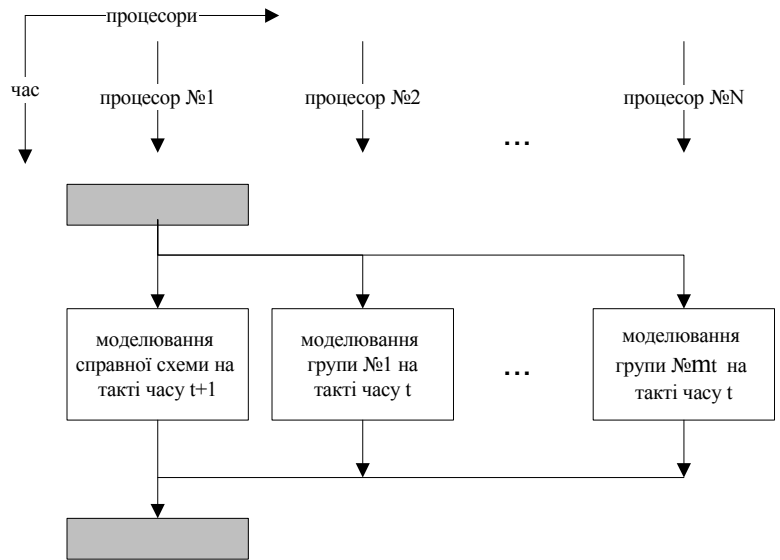


Рис.12. Діаграма потоків методу паралельного моделювання груп несправностей.

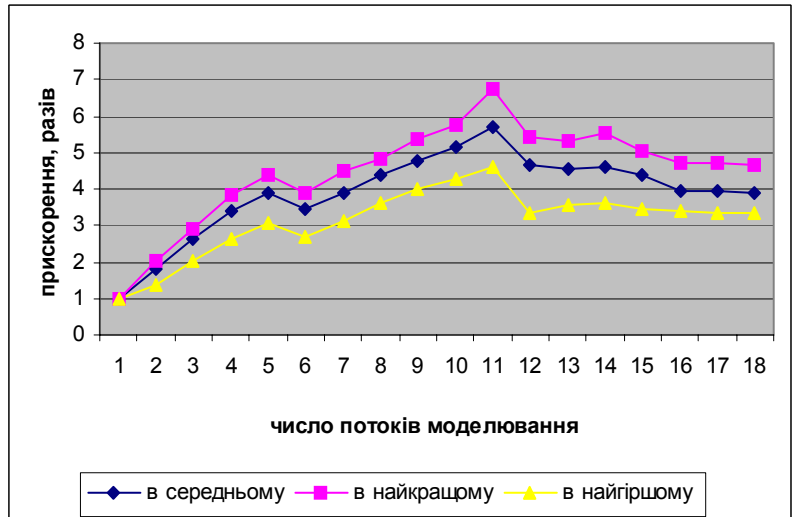


Рис.13. Прискорення роботи методу в залежності від числа потоків моделювання груп несправних ЦП.

При паралельному моделюванні на p процесорах з розбиванням списку несправностей отримуємо:

$$T_p(N, F, N_{эл}) = \gamma\beta N_{эл}N + \gamma\delta N_{эл}(N_F / p)L(1 - \alpha r). \quad (21)$$

Дана оцінка досягається в ідеальному випадку, коли не враховується швидкість передачі даних в мережі комунікації. З (21) випливає, що методи з розбиванням списку пошкоджень мають обмеження по масштабуванню при зростанні p через перший доданок, який показує, що поведінка справного ЦП має бути відновлена на всіх вузлах обчислювального кластера.

Для методу паралельного моделювання груп несправностей найбільше прискорення отримано для конфігурації, коли один вузол виконує моделювання справного ЦП, а $p-1$ вузол - моделювання несправних ЦП. Тоді:

$$T_p^1(L, N_F, N_{эл}) = \gamma\beta N_{эл}L;$$

$$T_p^i(L, N_F, N_{эл}) = \gamma\delta N_{эл}(N_F / (p-1))L(1 - \alpha r), \quad i = \overline{2, p}.$$

В T_p^i , що показує час моделювання несправних ЦП, вдалося позбавитися від немасштабованого доданка. І хоча для системи з p процесорами властивість масштабності буде обмежено підсистемою з $p-1$ процесорів, експерименти показали, що саме така конфігурація є найбільш ефективною.

У розділі 7 «Застосування алгоритмів симуляції відпалу в задачах діагностування цифрових пристроїв» методи діагностування та контролю узагальнені на непопулярній ЕА, зокрема - алгоритм СВ (рис.14). Основною його відмінністю від ГА є те, що тут відбувається еволюція одного потенційного рішення. Формально метод СВ задається як упорядкована сукупність об'єктів:

$$СВ = (K, R, C, T, K_0, N, l),$$

де: K - множина можливих конфігурацій (потенційних рішень); $R: K \rightarrow K$ - операція побудови конфігурацій в оточенні; $C: K \rightarrow R$ - функція вартості рішення; $T = \{T_1, T_2, \dots\}$ - розклад температур; $K_0 \in K$ - початкова конфігурація; $N(O(K_i)) = |O(K_i)|$ - розмір оточення для конфігурації $K_i \in K$; l - довжина кодування конфігурації в бітах.

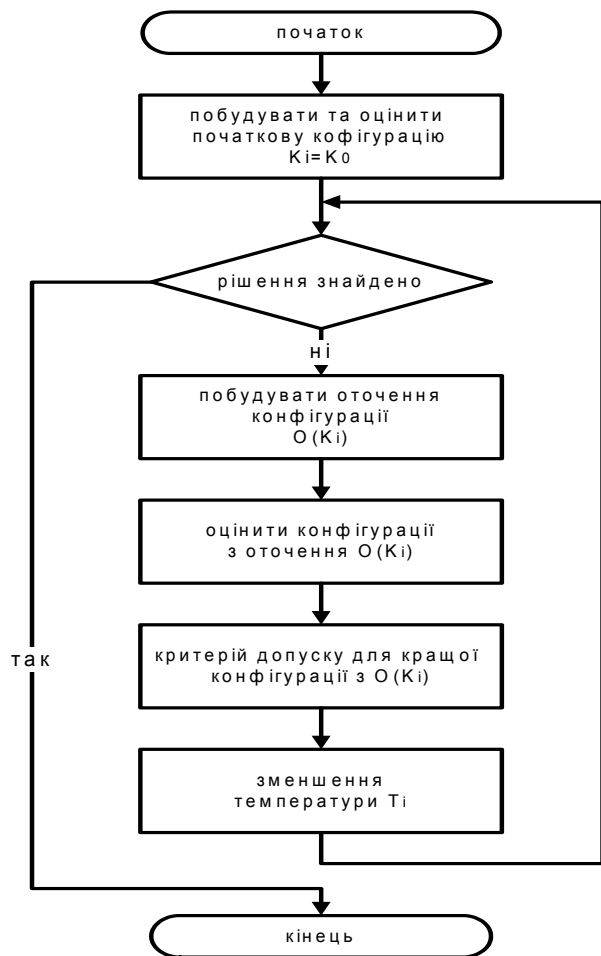


Рис.14. Схема алгоритму СВ.

Мета методу - пошук конфігурації з найвищою оцінкою: $C \rightarrow \max$.

Від методів найшвидшого спуску алгоритм СВ відрізняється імовірнісним механізмом побудови рішень та можливістю на окремих кроках приймати рішення з гіршою оцінкою. Механізм, що дозволяє приймати рішення, які погіршують оцінку, реалізується наступним чином. Для кожного значення температури з розкладу T виконується певна кількість збурюючих операцій R з породженням проміжної конфігурації $K_{пром} \in K$. У разі, якщо оцінка проміжної конфігурації-послідовності буде вища за оцінку поточної конфігурації $C(K_{пром}) > C(K_i)$, то вона замінює поточну конфігурацію: $K_i = K_{пром}$. В іншому випадку найгірша конфігурація може замінити поточну із певною ймовірністю $P(T_i, \Delta C)$, де $\Delta C = C(K_i) - C(K_{пром})$. Можливість погіршувати рішення, що дозволяє методу переходити в інші підпростори пошуку та уникати локальних оптимумів, відповідає мутації в ГА.

Застосовуючи аналогічний розглянутому в розділі 2 підхід, побудовано шаблон однорівневого СВ-методу генерації ІдП.

Далі в розділі розроблено однорівневі СВ-методи побудови послідовностей ініціалізації та верифікації еквівалентності поведінки ЦП, а також дворівневий СВ-метод побудови перевіряючих тестів на основі схеми з активізацією несправності. Близькість підходу до ГА дозволяє при конструктивній побудові методів запозичувати наступні компоненти: кодування рішень-послідовностей (рис.2), збурюючі операції, вигляд функцій оцінки.

За аналогічною схемою розробляється СВ-метод комбінаторної оптимізації з вибору субоптимальної підмножини в задачі побудови енергоефективних тестів (розділ 4).

При алгоритмічній реалізації СВ-методів також відбувається експериментальне обґрунтування евристичних параметрів: розкладу температур T , розподілу приймати рішення з меншою оцінкою, критеріїв зупинки тощо.

Порівняння ефективності парних ГА- та СВ-методів в термінах відповідних задач проведено в розділі 8.

У розділі 8 «Автоматизована система моделювання та діагностування АСМІД-Evolution» описано нову версію системи, основу якої складають еволюційні методи побудови та оптимізації ідентифікуючих послідовностей, запропоновані в даному дослідженні.

Основними компонентами системи та реалізованими в них функціями є наступні.

1. Блок попередньої обробки: введення текстового опису ЦП на логічному рівні та його трансляція у внутрішню структуру даних, представлена системою таблиць.

2. Блок моделювання: логічне моделювання поведінки непошкодженого ЦП на заданій вхідній послідовності в 3-х та 16-и значних алфавітах; логічне моделювання поведінки ЦП з несправностями на заданій вхідній послідовності з метою вивчення її діагностичних властивостей; для моделювання використовуються 3-х та 16-и значний алфавіти. У підсистемі застосовуються методи паралельного моделювання ЦП з несправностями, які запропоновані в розділі 6.

3. Підсистеми GA-Analyze та SA-Analyze: генерація вхідних ІдП різних класів (тестові для множини одиночних константних несправностей, ініціюючі послідовності, послідовності для верифікації еквівалентності двох заданих схем); генерація надлишкових тестів; оцінка пікового одно-, n -тактного та сталого розсіювання тепла заданого ЦП. У підсистемі застосовуються паралельні методи генерації перевіряючих послідовностей, які запропоновані в розділі 5.

4. Блок постобробки та перегляду результатів.

Далі в розділі описані застосовувані в системі структури даних, а також програмна реалізація запропонованих у роботі методів і компонент системи.

Роботу системи апробовано на схемах з міжнародного каталогу ISCAS-89. Наведено чисельні результати з реалізаціями методів розділів 2, 3 та 7, які показують високу ефективність у термінах відповідних задач. Додатково проведено порівняння ефективності для парних ГА-та СВ-методів побудови ІдП.

ВИСНОВКИ

У дисертаційній роботі запропоновано вирішення актуальної науково-практичної проблеми розробки єдиної методології побудови еволюційних моделей, методів і алгоритмів, в тому числі паралельних, які дозволяють вирішувати широкий клас задач діагностики ЦП великої розмірності. Запропонована методологія базується на двох основних положеннях: компонентний синтез нових еволюційних методів на підставі шаблонів еволюції рішень та широке застосування паралельних обчислень. Застосування еволюційних методів дозволяє змістити парадигму пошуку рішення з задачі синтезу на ітеративну задачу аналізу, що дозволяє проводити діагностування великих сучасних ЦП із значним енергоспоживанням. На практиці пропонується підхід дозволяє будувати сучасні програмні системи моделювання та діагностування цифрових пристроїв необхідного функціонального наповнення для паралельних обчислювальних систем різних класів. Проведені дослідження дозволяють зробити наступні висновки.

1. Запропоновано дві моделі застосування ЕА в задачах побудови ідентифікуючих послідовностей ЦП: одно- і дворівневі, в яких крім основної частини методу виділяються залежні від реалізації компоненти, кінцевий вигляд яких або їх числові значення визначаються на підставі машинних експериментів. Це дозволяє на основі таких моделей шляхом визначення оцінюючої функції, параметрів еволюції і залежних від реалізації компонент синтезувати цілий ряд нових ефективних методів побудови ідентифікуючих послідовностей ЦП, зменшуючи час розробки та реалізації таких методів.

2. На основі однорівневої моделі розроблено ГА-методи побудови ідентифікуючих послідовностей ЦП: логічної ініціалізації, досягнення стану, верифікації еквівалентності. Дані методи дозволяють ефективно вирішувати зазначені задачі для пристроїв великої розмірності в термінах числа ініційованих тригерів, довжини послідовності, якості верифікації та часу роботи.

3. На основі дворівневої моделі розроблено ГА-методи побудови перевіряючих, діагностичних і функціональних тестів, які дозволяють ефективно вирішувати дані завдання для ЦП великої розмірності в термінах часу роботи і досягнутої повноти. Показано, що на нижньому рівні використовуються відповідні

однорівневі ГА-методи.

4. Вперше введено поняття функцій поведінки, активності та відмінності компонент за парою та множиною ЦП. Дані функції, з одного боку, відображають поведінку пристрою на структурному рівні, а з іншого - визначають ту додаткову інформацію, яка дає перевагу структурних методів діагностування над абстрактними. Показано, що всі динамічні параметри оцінюючих функцій виражаються через дані функції.

5. Розроблено новий метод діагностичного моделювання великих ЦП, який призначений для роботи спільно з ГА-методом генерації діагностичних послідовностей та дозволяє обчислювати відомі діагностичні міри.

6. Розроблено ГА-методи оцінки розсіювання тепла ЦП для заданих режимів роботи: пікове одноктне, n -тактне та стале. Показано зв'язок між даними параметрами. Рішення задач оцінки зведено до однорівневих ГА-методів побудови ідентифікуючих послідовностей з урахуванням передпочаткового стану і послідовності досягнутих станів ЦП. Розроблено підхід побудови енергоефективних тестів, в рамках якого розроблено ГА-методи надлишкового тестування і вибору субоптимальної підмножини тестів.

7. Розроблено методику побудови паралельних версій ГА-методів за схемою «господар-робітник», в якій використовуються різні схеми паралельного моделювання цифрових пристроїв в процедурах оцінки особин-послідовностей. Розроблено нові ПГА для слабо- та сильнопаралельних ОС, досліджено характеристики їх масштабованості. Швидкодію методів генерації послідовностей збільшено в 1.6-2.2 рази на двоядерній системі, в 1.1-3.2 рази на 4-ядерній системі, в 2.2-13.5 рази на 12-ядерній системі.

8. Для схеми «острівів» ПГА розроблено нові методи роботи компонент сервера та клієнтів. Особливістю запропонованого підходу є централізоване управління сервером усіма структурними компонентами, що дозволяє реалізовувати довільні топології взаємодії островів і стратегії адаптації їх параметрів. При цьому за рахунок варіації даних параметрів фактично реалізується побудова нових ПГА-методів для доступного класу паралельних обчислювальних систем.

9. На підставі схеми з розбиттям списку несправностей розроблено методи паралельного моделювання для ОС з розподіленою та загальною пам'яттю. Швидкодію методів моделювання з пошкодженнями підвищено в 4.4-7.0 разів на 8-процесорному кластері та в 2.8-3.4 рази на 4-ядерній системі. Для багатоядерної ОС із загальною пам'яттю вперше запропоновано принципово новий метод, який полягає в паралельному моделюванні груп несправностей, що дозволило підвищити швидкість в 4.6-6.7 рази на 12-ядерній системі. Сукупність розроблених методів дозволяє адаптувати ЕА, в яких застосовується моделювання ЦП з пошкодженнями, на паралельні апаратні платформи різних класів.

10. Запропонований підхід побудови ЕА діагностування ЦП узагальнено на методи з еволюцією одного рішення. Зокрема, на підставі алгоритму симуляції відпалу розроблено ряд однорівневих методів побудови ідентифікуючих послідовностей ЦП, дворівневий метод генерації перевіряючих тестів ЦП, метод обирання субоптимальної підмножини в задачі побудови енергоефективних тестів. При цьому запропоновані методи використовують компоненти генетичних алгоритмів: коду-

вання рішень, еволюційні операції, оцінюючі функції.

11. На підставі програмної реалізації запропонованих у роботі методів та підходів виконано побудову нової версії системи моделювання та діагностики ЦП «АСМІД-Evolution». Застосування нових ЕА в задачах пошуку рішень дозволяє підвищити експлуатаційні характеристики системи за рахунок обробки ЦП великої розмірності та будувати тестові послідовності з поліпшеними характеристиками: повноту перевіряючих тестів підвищено на 7-10%, ступінь верифікації підвищено на 12-15%, розсіювання тепла в процесі тестування знижено на 75-93%. Розроблена система знайшла застосування на виробництві при проектуванні та діагностуванні цифрової апаратури, а також у навчальному процесі.

СПИСОК ОСНОВНИХ ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Иванов Д.Е. Генетические алгоритмы построения входных идентифицирующих последовательностей цифровых устройств / Д.Е. Иванов. – Донецк, 2012. – 240с.
2. Иванов Д.Е. Методы параллельного моделирования СБИС с неисправностями / Д.Е. Иванов // Радиоэлектронні і комп'ютерні системи, 2012.- №5.- С.114-119.
3. Иванов Д.Е. Оценка рассеивания тепла СБИС для различных режимов работы / Д.Е. Иванов // Вісник східноукраїнського національного університету ім.В.Даля, 2012.- №8(179).- С.44-51.
4. Иванов Д.Е. Алгоритмы достижения состояний в цифровых устройствах и их применение в задачах диагностики / Д.Е. Иванов // Вісник Хмельницького національного університету. Технічні науки.- Хмельницький, 2012.- №3(189).- С.104-110.
5. Иванов Д.Е. Алгоритм диагностического моделирования СБИС / Д.Е. Иванов // Проблемы информационных технологий.- 2012.- №1 (011).- С.107-114.
6. Иванов Д.Е. Параллельный алгоритм моделирования цифровых схем с неисправностями для многоядерных систем с общей памятью / Д.Е. Иванов // Электронное моделирование.- 2011.- Т.33, №2.- С.93-106.
7. Иванов Д.Е. Масштабируемый параллельный генетический алгоритм построения идентифицирующих последовательностей для современных многоядерных вычислительных систем / Д.Е. Иванов // Управляющие системы и машины.- 2011.- №1.- С.25-32.
8. Иванов Д.Е. Автоматизированная система моделирования и идентификации цифровых устройств АСМІД- Evolution / Д.Е. Иванов // Проблемы информационных технологий.- 2011.- №1 (009).- С.114-124.
9. Иванов Д.Е. Генетический алгоритм оценки пикового рассеивания тепла для больших интегральных схем / Д.Е. Иванов, Т.А. Васяева // Вісник східноукраїнського національного університету ім. В.Даля, 2011.- №13(167).- С.277-283.
10. Иванов Д.Е. Применение алгоритмов симуляции отжига в задачах идентификации цифровых схем / Д.Е. Иванов // Вісник Національного технічного університету "Харківський політехнічний інститут". Збірник наукових праць. Тематичний випуск: Інформатика і моделювання.– Харків: НТУ "ХПІ", 2011.– № 17.– С.60-69.

11.Иванов Д.Е. Подходы к построению параллельных генетических алгоритмов идентификации цифровых схем для многоядерных систем / Д.Е. Иванов // Вісник Хмельницького національного університету. Технічні науки.- Хмельницький, 2011.- №1(172).- С.111-117.

12.Иванов Д.Е. Генетический алгоритм построения диагностических последовательностей цифровых устройств / Д.Е. Иванов // Вісник східноукраїнського національного університету ім. В.Даля, 2010.- №10(152).- С.72-79.

13.Иванов Д.Е. Генетический алгоритм оптимизации рассеивания тепловой энергии входных тестовых последовательностей / Д.Е. Иванов // Наукові праці Донецького національного технічного університету. Серія: “Обчислювальна техніка та автоматизація”.- Донецьк:ДонНТУ, 2010.- №18(169).- С.206-215.

14.Иванов Д.Е. Алгоритм симуляции отжига построения тестов цифровых устройств / Д.Е. Иванов, Р. Зуауи // Вестник Херсонского национального технического университета.- 2010.- №2(38).- С.416-422.

15.Иванов Д.Е. Алгоритм симуляции отжига оптимизации рассеивания тепла диагностических тестов / Д.Е. Иванов, Р. Зуауи // Радіоелектронні і комп'ютерні системи.- 2010.- №7(48).- С.170-175.

16.Иванов Д.Е. Алгоритм построения инициализирующих последовательностей цифровых схем, основанный на стратегии симуляции отжига / Д.Е. Иванов, Р. Зуауи // Искусственный интеллект.- 2009.- №4.- С.415-424.

17.Иванов Д.Е. Верификация эквивалентности цифровых схем с использованием стратегии симуляции отжига / Д.Е. Иванов, Р. Зуауи // Науковий вісник Чернівецького університету. Комп'ютерні системи та компоненти.- 2009.- №479.- С.33-41.

18.Иванов Д.Е. Алгоритм параллельного вычисления оценок особей при верификации эквивалентности последовательностных схем / Д.Е. Иванов // Проблемы информационных технологий.- 2009.- №1(005).- С.105-112.

19.Ivanov D.E. Parallel fault simulation on multi-core processors / D.E. Ivanov // Радіоелектронні і комп'ютерні системи.- 2009.- №6(40).- С.109-112.

20.Иванов Д.Е. Генетический подход проверки эквивалентности последовательностных схем / Д.Е. Иванов // Радіоелектроніка. Інформатика. Управління.- Запоріжжя, ЗНТУ.- 2009.- №1(20).- С.118-123.

21.Иванов Д.Е. Параллельный генетический алгоритм верификации эквивалентности цифровых схем для двухядерных систем / Д.Е. Иванов // Вісник Хмельницького національного університету.- 2009.- №4.- С.92-99.

22.Иванов Д.Е. Взаимодействие компонент в распределённых генетических алгоритмах генерации тестов / Д.Е. Иванов, П.А. Чебанов // Наукові праці Донецького національного технічного університету. Серія: “Обчислювальна техніка та автоматизація”.- Донецьк:ДонНТУ, 2009.- №16(147).- С.121-127.

23.Иванов Д.Е. Применение стратегии симуляции отжига для задачи построения инициализирующих последовательностей цифровых схем / Д.Е. Иванов, Р. Зуауи // Вісник східноукраїнського національного університету ім.В.Даля.- 2009.- №1(131), частина 2.- С.161-168.

24.Иванов Д.Е. Генетические алгоритмы построения идентифицирующих последовательностей для цифровых схем с памятью / Д.Е. Иванов // Наукові праці До-

нецького національного технічного університету. Серія: “Обчислювальна техніка та автоматизація”.- Донецьк:ДонНТУ, 2008.- №14(129).- С.97-106.

25.Иванов Д.Е. Построение инициализирующих последовательностей синхронных цифровых схем с помощью генетических алгоритмов / Д.Е. Иванов, Ю.А. Скобцов, А.И. Эль-Хатиб // Проблеми інформаційних технологій.- 2007.- №1.- С.158-164.

26.Иванов Д.Е. Распределённые генетические алгоритмы генерации проверяющих тестов цифровых систем / Д.Е. Иванов, Ю.А. Скобцов, А.И. Эль-Хатиб // Радіоелектронні комп'ютерні системи.- ХАІ, 2007.- №7.- С.176-181.

27.Д.Е. Иванов Распределённые алгоритмы моделирования и генерации тестов / Д.Е. Иванов, Ю.А. Скобцов, А.И. Эль-Хатиб. // Радіоелектронні і комп'ютерні системи.-ХАІ, 2006.- №6.- С.97-102.

28.Иванов Д.Е. Распределённое параллельное моделирование цифровых схем с неисправностями / Д.Е. Иванов, Ю.А. Скобцов, А.И. Эль-Хатиб // Наукові праці Донецького національного технічного університету. Серія: “Обчислювальна техніка та автоматизація”.- Донецьк:ДонНТУ, 2006.- №107.- С.128-134.

29.Иванов Д.Е. Исследование влияния параметров генетического алгоритма при генерации тестов для последовательностных схем / Д.Е. Иванов, Ю.А. Скобцов, П.А. Чебанов // Вісник Донецького університету, Сер.А: Природничі науки.- Донецьк, 2005.- №2.- С.397-402.

30.Иванов Д.Е. Эволюционный подход к функциональному тестированию цифровых схем / Д.Е. Иванов, Ю.А. Скобцов // Наукові праці Донецького національного технічного університету. Серія: “Обчислювальна техніка та автоматизація”.- Донецьк:ДонНТУ, 2004.- №74.- С.135-140.

31.Иванов Д.Є. Еволюційні методи побудови перевіряючих тестів для цифрових схем / Д.Є. Иванов, Ю.О. Скобцов, В.Ю. Скобцов // Вісник технологічного університету Поділля.- Хмельницький, 2004.- С.135-139.

32.Ivanov D.E. Evolutionary approach to the test pattern generation for the sequential circuits / Y.A. Skobtsov, D.E. Ivanov // Радиоэлектроника и информатика.- 2003.- №3.- С.46-51.

33.Иванов Д.Е. Применение адаптивных генетических алгоритмов для генерации тестов цифровых схем / Д.Е. Иванов, Ю.А. Скобцов, В.Ю. Скобцов, С.А. Закусило // Наукові праці Донецького національного технічного університету, серія “Обчислювальна техніка та автоматизація».- Донецьк, 2002.- №47.- С.249-255.

34.Иванов Д.Е. Генетические алгоритмы в диагностике и проектировании цифровых схем / Д.Е. Иванов, Ю.А. Скобцов, В.Ю. Скобцов // Искусственный интеллект.- 2002.- №2.- С.250-258.

35.Иванов Д.Є. Адаптивні механізми в генетичних алгоритмах / Д.Є. Иванов, Ю.О. Скобцов, С.А. Закусило // Наукові праці Донецького державного технічного університету, серія “Обчислювальна техніка та автоматизація».- Донецьк, 2002.- №38.- С.104-109.

36.Иванов Д.Е. Генетические алгоритмы в генерации проверяющих тестов цифровых систем / Д.Е. Иванов, В.Ю. Скобцов, Ю.А. Скобцов // Інформаційно-керуючі системи на залізничному транспорті.- 2001.- №4.- С.52-55.

37.Иванов Д.Е. Сжатие списка неисправностей с помощью генетического алго-

ритма / Д.Е. Иванов, В.Ю. Скобцов // Наукові праці Донецького державного технічного університету, серія “Обчислювальна техніка та автоматизація».- Донецьк, 2001.- №25.- С.161-167.

38.Иванов Д.Е., Скобцов Ю.А. Ускорение работы генетических алгоритмов при построении тестов / Д.Е. Иванов, Ю.А. Скобцов // Искусственный интеллект.- 2001.- №1.- С.52-60.

39.Ivanov D. Development of methods of parallel simulation in evolutionary algorithms of identification of VLSI / D. Ivanov // Book of abstracts of 20th Conference on applied and industrial mathematics CAIM 2012.- Iasi University press, 2012.- P.58.

40.Ivanov D. A scalable genetic algorithm for constructing of identifying sequences for VLSI circuits / D. Ivanov // Book of abstracts of 19th Conference on applied and industrial mathematics CAIM 2011.- Iasi University press, 2011.- P.68-69.

41.Ivanov D. Parallel fault simulation algorithm of digital circuits for many-core workstations with common memory / D. Ivanov // Book of abstracts of 18th Conference on applied and industrial mathematics CAIM 2010.- Iasi University press, 2010.- P.46.

42.Ivanov D.E. Evolutionary distributed test generation methods for digital circuits / D.E. Ivanov, Y.A. Skobtsov, V.Y. Skobtsov // Proc. of 8th International Workshop on Boolean Problems, September 18-19, 2008.- Freiberg, Germany, 2008.- P.213-218.

43.Ivanov D.E. Distributed Fault Simulation and Genetic Test Generation of Digital Circuits / D.E. Ivanov, Y.A. Skobtsov, A.I. El-Khatib // Proceedings of IEEE East-West Design & Test Workshop (EWDT'06).- Sochi, 2006.- P.89-94.

44.Ivanov D.E. Distributed Genetic Algorithm of Test Generation For Digital Circuits / D.E. Ivanov, Y.A. Skobtsov, A.I. El-Khatib // Proceedings of the 10th Biennial Baltic Electronics Conference.- Tallinn Technical University, 2006.- P.281-284.

45.Ivanov D.E. Parallel Genetic Algorithm of Test Generation for Digital Circuits / D.E. Ivanov, Y.A. Skobtsov, El-Khatib // Proceedings of the International Conference “Modern problems of Radio Engineering, Telecommunications and Computer Science”.- Lviv-Slavsko, 2006.- P.129-131.

46.Ivanov D.E. Evolutionary Approach to Test Generation for Functional BIST / D.E. Ivanov, Y.A. Skobtsov, V.Y. Skobtsov, R. Ubar, J.Raik // 10 European Test Symposium. Informal Digest of Papers.- 2005.- P.151-155.

47.Ivanov D.E. Evolutionary approach to the functional test generation for digital circuits / D.E. Ivanov, Y.A. Skobtsov, V.Y. Skobtsov, R. Ubar // Proc. of 9th Biennial Baltic Electronics Conf., BEC 2004.- Tallinn Univ. of Techn., 2004.- P.229-232.

48.Ivanov D.E. Genetic algorithms in test generation for digital circuits / D.E. Ivanov, Y.A. Skobtsov, V.Y. Skobtsov // Proceedings of the 8th Biennial Baltic Electronics Conference.-Tallinn Technical University, 2002.- P.291-294.

АНОТАЦІЯ

Іванов Д.Є. Діагностування цифрових пристроїв на базі еволюційного підходу.- На правах рукопису.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.05 – Комп’ютерні системи та компоненти.- ДВНЗ «Донецький національний технічний університет», Донецьк, 2013.

Дисертацію присвячено розробці методологічних основ еволюційного підходу

до діагностування цифрових пристроїв. Даний підхід включає набір еволюційних моделей, методів, алгоритмів та евристик для побудови ідентифікуючих послідовностей широкого класу та їх оптимізації, які здатні обробляти сучасні цифрові пристрої великої розмірності та призначені для реалізації на сучасних паралельних обчислювальних системах. Підхід розповсюджено як на популяційні методи (генетичний алгоритм), так і на методи з еволюцією одного потенційного рішення (алгоритм симуляції відпалювання).

Для оцінки особин в таких методах виконується моделювання цифрового пристрою (справного або з пошкодженнями) на заданій послідовності. Для формалізації побудови оцінок особин-послідовностей введено функції поведінки та активності компонент, відмінності за парою та множиною пристроїв. Ці функції відображають інформацію щодо поведінки цифрових пристроїв на структурному рівні, яка дає перевагу над абстрактними методами.

З метою адаптації еволюційних методів до сучасних паралельних систем різноманітних класів розроблено паралельні версії генетичних алгоритмів за схемами «хазяїн-робітник» та «острів», що мають за мету прискорення процесу пошуку та його якості відповідно, а також паралельні методи моделювання цифрових пристроїв з пошкодженнями.

На основі запропонованих методів та алгоритмів розроблено сучасну систему моделювання та ідентифікації АСМІД-Evolution.

Ключові слова: цифровий пристрій, діагностика, еволюційні алгоритми, розсіювання тепла, паралельні обчислення, моделювання з пошкодженнями.

АННОТАЦІЯ

Иванов Д.Е. Диагностирование цифровых устройств на базе эволюционного подхода.- На правах рукописи.

Диссертация на соискание учёной степени доктора технических наук по специальности 05.13.05 – Компьютерные системы и компоненты.- ГВУЗ «Донецкий национальный технический университет», Донецк, 2013.

Диссертация посвящена разработке методологических основ эволюционного подхода в диагностировании цифровых устройств. Данный подход включает набор эволюционных моделей, методов, алгоритмов и эвристик для построения идентифицирующих последовательностей широкого класса, а также их оптимизации, которые способны проводить обработку устройств большой размерности и предназначены для реализации на параллельных вычислительных системах широкого класса. Подход распространен как на популяционные методы (генетический алгоритм), так и на методы с эволюцией одного потенциального решения (алгоритм симуляции отжига).

Предложено две модели применения эволюционных алгоритмов в задачах построения идентифицирующих последовательностей: одно- и двухуровневая. В одноуровневой модели единственный цикл эволюции потенциального решения формирует метод решения задачи. В двухуровневой модели происходит итеративный выбор промежуточных целей и их достижения с помощью эволюционного алгоритма, что ведёт к решению общей задачи.

На основе одноуровневой модели разработаны эволюционные методы решения задач логической инициализации, верификации эквивалентности поведения двух за-

данных устройств и задачи достижения заданного состояния в устройстве.

На основе двухуровневой (итерационной) модели разработаны эволюционные методы решения задач построения проверяющих тестов (на основе активизации неисправностей и с подтверждением состояний) и диагностических тестов, задачи тестирования функциональных блоков арифметико-логических устройств.

Для формализации оценок особей-последовательностей введены понятия функций поведения и активности компонент, функции различия компонент по паре и множеству устройств. Показано, что через данные функции выражаются все динамические параметры оценочных функций особей в разрабатываемых эволюционных методах.

Для задач, возникающих при проектировании энергоэффективных устройств, разработаны эволюционные методы оценки рассеивания тепла в зависимости от режимов работы, которые основаны на генетических алгоритмах. Для задачи стартового тестирования разработан подход построения энергоэффективных тестов, позволяющий снизить рассеивание тепла во время такого тестирования. В данном подходе этапы построения избыточных тестов и выбора субоптимального подмножества тестовых последовательностей решены с помощью генетического алгоритма.

С целью адаптации предлагаемых методов на параллельные вычислительные системы различной структуры разработаны параллельные генетические алгоритмы, основанные на схемах «хозяин-рабочий» и «островов», которые предназначены как для ускорения работы методов, так и для повышения качества поиска решений. Методы класса «хозяин-рабочий» используют различные схемы параллельного моделирования цифровых устройств в процедурах оценки особей-последовательностей. Для схемы «островов» параллельных генетических алгоритмов разработаны методы, реализующие работу распределённых компонент сервера и клиентов, что позволяет выполнять централизованное управление процессом миграции, реализовывать различные топологии их взаимодействия и адаптации параметров эволюции.

Разработан ряд методов моделирования цифровых устройств с неисправностями для параллельных вычислительных систем с общей и распределённой памятью, которые позволяют адаптировать эволюционные алгоритмы для работы на таких системах, а также повысить скорость их работы. Предложен принципиально новый метод параллельного моделирования цифровых устройств с неисправностями для многоядерных вычислительных систем с общей памятью. Данный метод использует технику параллельного моделирования групп неисправностей для каждого входного набора из тестовой последовательности.

Предлагаемая методология распространена на методы с эволюцией одного решения – алгоритм симуляции отжига. Это позволило разработать соответствующие методы построения идентифицирующих последовательностей, при синтезе которых часть компонент заимствуется из генетических алгоритмов.

Практическое значение данной работы заключается в том, что на основе разработанных в работе методов построена новая система моделирования и диагностики АСМИД-Evolution. Разработанный единый подход позволяет конструктивно выполнять синтез целого ряда новых эволюционных методов генерации идентифицирующих последовательностей различных типов, которые могут быть реализованы на современных параллельных инструментальных платформах широкого класса. Практи-

чески показано применение ряда эвристик, позволяющих повысить эффективность разрабатываемых эволюционных методов.

Основные результаты нашли применение на производстве при проектировании цифровой аппаратуры и цифровых компонент гибридной аппаратуры, а также в учебном процессе.

Ключевые слова: цифровое устройство, диагностика, эволюционные алгоритмы, рассеивание тепла, параллельные вычисления, моделирование с неисправностями.

ABSTRACT

Ivanov D.E. Diagnosis of digital devices based on an evolutionary approach.- As the manuscript.

Thesis for a doctor's degree in technical science by specialty 05.13.05 – Computer systems and components.– Donetsk National Technical University, Donetsk, 2013.

Dissertation is devoted to developing of the methodological bases of the evolutionary approach of the diagnostics of digital devices. This approach includes a set of evolutionary models, methods, algorithms and heuristics for constructing of identifying sequences of the wide classes and for its optimization, which can handle modern digital devices of large size. Developed methods are designed for implementation on modern parallel computer systems. The approach expanded both on the population based methods (genetic algorithm) and on the methods with the evolution of one potential solution (simulated annealing algorithm).

To evaluate the individuals in such methods the simulation (fault or fault-free) on the given sequence is performed. To formalize the construction of estimation functions of sequences-individuals the functions of state and activity component, differences on a pair and on set devices are introduced. These functions express the information about the behavior of digital devices on the structural level, which gives the advantages over abstract methods.

In order to adapt evolutionary techniques to modern parallel systems of different classes the parallel versions of genetic algorithms with schemes "master-worker", "islands" and parallel methods of fault simulation of digital devices are developed. They are aimed to accelerate the search process and its quality.

On the basing on the proposed methods and algorithms the modern system of simulation and identification ASMID-Evolution is developed.

Keywords: digital device, diagnostics, evolutionary algorithms, heat dissipation, parallel computing, fault simulation.