

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)
(рівень вищої освіти)

Синтез схем діагностування несправності вхідної змінної
(тема)

Виконав: здобувач IV курсу, групи КІУКІ-21-9
Дорошин М. С.
(прізвище, ініціали)


Спеціальність 123 Комп'ютерна інженерія
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп'ютерна інженерія
(повна назва освітньої програми)

Керівник Хаханов В.І.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри АПОТ


(підпис)

Чумаченко С.В.
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія
(шифр і назва)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри АПОТ



Чумаченко С.В.

(підпис)

« 06 » 05 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Дорошину Михайлу Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Синтез схем діагностування несправності вхідної змінної
затверджена наказом по університету від " 21 " 05 2025 р. № 403 Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 15.06.2025

3. Вихідні дані до роботи (проекту)

Моделі, метрики, архітектури векторно-логічного комп'ютингу

Математичний апарат дискретної математики: таблиці істинності, булеві похідні

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

Огляд технологій

Аналіз моделей, метрик математичного апарату векторної логіки

Розробка схем діагностування несправностей вхідної змінної

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

21 слайд

6. Консультанти розділів роботи (проекту)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Видача теми проекту, узгодження і затвердження	06.05.2025 - 06.05.2025	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів. Складання аналітичного огляду стану технологій. Аналіз останніх досліджень та публікацій.	07.05.2025 - 10.05.2025	
3	Опис обраних моделей, методів та алгоритмів	10.05.2025 - 17.05.2025	
4	Розробка схем діагностики несправностей вхідної змінної	18.05.2025 - 23.05.2025	
5	Повна схема діагностування	24.05.2025 - 30.05.2025	
6	Верифікація	01.06.2025 - 05.06.2025	
7	Оформлення пояснювальної записки	06.06.2025 - 10.06.2025	
8	Перевірка виконаного проекту керівником, допуск до захисту	11.06.2025 - 12.06.2025	
9	Захист проекту	17.06.2025	

Дата видачі завдання 06.05.2025

Здобувач _____

(підпис)

Дорошин М. С.

Керівник роботи _____

(підпис)

проф. Хаханов В.І.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 45 с., 13 рисунків, 1 дод., 13 джерел.

СХЕМА, СИНТЕЗ, ВЕКТОРНО-ЛОГІЧНИЙ КОМП'ЮТИНГ, МЕТРИКА, КАРТА ТЕСТУВАННЯ, ДІАГНОСТУВАННЯ, НЕСПРАВНІСТЬ

Тематика роботи стосується питань синтезу схем діагностики несправності вхідної змінної на основі моделей, метрик та технологій векторно-логічного комп'ютингу.

Метою бакалаврської роботи є підвищення ефективності виявлення несправностей при проектуванні цифрових систем шляхом синтезу схем діагностики несправності вхідної змінної на основі векторно-логічних обчислювальних механізмів.

Задачі: огляд стану технологій; аналіз публікацій, дослідження моделей, методів, технологій векторно-логічного комп'ютингу, синтез схем діагностики несправності вхідної змінної.

ABSTRACT

Bachelor's thesis contains 45 pages format A4, 13 figures, 1 application, 13 sources.

CIRCUIT, SYNTHESIS, VECTOR LOGIC COMPUTING, METRICS, TESTING MAP, DIAGNOSTICS, FAULT

The topic of the work concerns the synthesis of input variable fault diagnosis circuits based on models, metrics and technologies of vector logic computing.

The purpose of the bachelor's thesis is to increase the efficiency of fault detection in the design of logic circuits by synthesizing input variable fault diagnosis circuits based on vector logic computing mechanisms.

Tasks: review of the state of technologies; analysis of publications, research of models, methods, technologies of vector logic computing, synthesis of input variable fault diagnosis circuits.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
1 СТАН ТЕХНОЛОГІЙ	9
1.1 Топ тренди Gartner 2025	9
1.2 Аналіз літератури	13
1.3 Висновки та постановка завдання	20
2 МАТЕМАТИЧНИЙ АПАРАТ ТА ТЕХНОЛОГІЧНІ РІШЕННЯ ВЕКТОРНО-ЛОГІЧНОГО КОМП'ЮТИНГУ	22
2.1 Визначення комп'ютингу	22
2.2 Design and Test комп'ютинг	24
2.3 Розумні структури даних	24
2.4 Механізм адресного заповнення Fault Simulation Matrix	27
2.5 Таблиця істинності для моделювання несправностей	28
2.6 Апарат булевих похідних для синтезу тестів функціональності за логічним вектором	30
2.7 Висновки до розділу 2	33
3 СИНТЕЗ СХЕМ ДІАГНОСТИКИ НЕСПРАВНОСТЕЙ ВХІДНОЇ ЗМІННОЇ	34
3.1 Використання булевих похідних для тестування цифрових схем	34
3.2 Визначення несуттєвості вхідної змінної за булевою похідною	35
3.3 Повна схема діагностування логічної функціональності	41
3.4 Висновки до розділу 3	44
ВИСНОВКИ	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	46
ДОДАТОК А	48
Графічна частина	48

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence (штучний інтелект);

ASIC – спеціалізована інтегральна схема;

CPU – центральний процесор;

FAAS – Faults-as-address circuit simulation;

FPGA – програмована вентиляна матриця;

GenAI – Generative AI (генеративний ШІ);

GPU – графічний процесор;

IoT – Internet of Things;

PQC – post-quantum cryptography (постквантова криптографія);

ДНФ – диз'юнктивна нормальна форма;

ШІ – штучний інтелект.

ВСТУП

Тематика роботи стосується питань синтезу схем діагностики несправності вхідної змінної на основі моделей, метрик та технологій векторно-логічного комп'ютингу.

Метою бакалаврської роботи є підвищення ефективності виявлення несправностей при проектуванні цифрових систем шляхом синтезу схем діагностики несправності вхідної змінної на основі векторно-логічних обчислювальних механізмів.

Задачі: огляд стану технологій; аналіз публікацій, дослідження моделей, методів, технологій векторно-логічного комп'ютингу, синтез схем діагностики несправності вхідної змінної.

1 СТАН ТЕХНОЛОГІЙ

Аналізується стан сучасних технологій на основі прогнозів консалтингової компанії Gartner на 2025 рік. Наводиться огляд літератури.

1.1 Топ тренди Gartner 2025

Стратегічні технологічні тренди визначають майбутнє, стимулюючи інновації [1,2]. Водночас вони порушують питання етики, відповідальності та довіри. Лідерські якості керівників ІТ перевіряються їхньою здатністю передбачати майбутнє та підготуватися до нього, та огляд стратегічних технологічних трендів від Gartner – компас на цьому шляху.

Gartner ретельно відібрав 10 головних трендів виходячи з їхнього потенціалу руйнування традиційних бізнес-моделей, стимулювання інновацій та здатності відповісти на найбільш гострі питання, що стоять сьогодні перед бізнесом. Це стратегічні тренди, що вимагають вдумливого розгляду та рішучих дій.

Розглянемо у подробицях кожен технологічний тренд та сценарії їх використання сьогодні та завтра.

1. Агентський ШІ (agentic AI) – це програмне забезпечення, здатне самостійно приймати рішення та виконувати дії, спрямовані на досягнення певних цілей. Для цього він використовує такі функції ШІ, як запам'ятовування, планування, сприйняття навколишнього середовища, використання інструментів та дотримання правил безпеки.

Прогноз Gartner: частка повсякденних рішень, що приймаються автономним агентським ШІ, зросте з нуля у 2024 році до 15% у 2028 році.

«Організації постійно прагнуть підвищувати ефективність, покращувати крос-функціональну взаємодію та координувати дії команд. Агентський ШІ має потенціал висококомпетентного цифрового

співробітника, здатного виявляти в закономірності, що відбувається, не помітні співробітникам-людям.» – Том Кошоу (Tom Coshou), Gartner.

2. Платформи регулювання ШІ (AI governance platforms) забезпечують відповідальне та етичне використання систем на базі ШІ. Вони дають ІТ-керівникам впевненість у тому, що ШІ надійний, прозорий, чесний, підконтрольний, відповідає стандартам етики та безпеки. Така платформа – гарантія того, що ШІ не суперечить цінностям організації та очікуванням суспільства загалом.

Прогноз Gartner: до 2028 року корпорації, що застосовують платформи регулювання ШІ, отримають рейтинги довіри споживачів на 30% вищі, ніж у конкурентів, а рейтинги відповідності регуляторним вимогам – на 25% вищі.

«Коли ми використовуємо банківську програму або сайт, ми часто, самі того не знаючи, стикаємося з ШІ, який схвалює заявку на кредит, відстежує можливе шахрайство або дає персоналізовану фінансову раду. Платформа регулювання ШІ допомагає банку забезпечити справедливість та етичність прийнятих рішень, захист персональних даних та дотримання вимог регулятора.» – Джаслін Каур Сінду (Jasleen Kaur Sindhu), Gartner.

3. Захист від дезінформації. Технології захисту від дезінформації повинні показувати, чому можна вірити. Вони націлені на створення систем, що забезпечують точність та автентичність інформації, що запобігають підробці та відстежують поширення шкідливого контенту.

Прогноз Gartner: до 2028 року продукти, послуги або можливості запобігання дезінформації використовуватимуть 50% корпорацій порівняно з 5% у 2024 році.

«Уявіть собі, що співробітник отримує від генерального директора листа, в якому запитується конфіденційна інформація або потрібно підтвердити проведення фінансової транзакції. Технології боротьби з дезінформацією проаналізують вміст листа, метадані та його походження і виявлять можливу фальсифікацію або шахрайство. При необхідності лист

автоматично потрапить до карантину, а співробітник та служба безпеки отримають відповідні повідомлення.» – Ден Аюб (Dan Ayoub), Gartner.

4. Пост-квантова криптографія. Постквантова криптографія (post-quantum cryptography, PQC) – це методи шифрування, спроектовані так, щоб бути захищеними від потенційних загроз з боку квантових комп'ютерів.

Прогноз Gartner: до 2029 року прогрес у галузі квантових обчислень зробить більшість загальноприйнятих методів асиметричного шифрування небезпечними.

«Алгоритми постквантової криптографії можна застосовувати, коли співробітники пересилають електронною поштою конфіденційну інформацію, що стосується клієнтів або інтелектуальної власності. Навіть якщо зломисники зараз перехоплять дані, вони не зможуть розшифрувати їх у майбутньому, коли квантові комп'ютери стануть достатньо потужними, щоб зламати існуючі стандартні методи шифрування». – Марк Хорват (Mark Horvath), Gartner.

5. Фоновий ІІІ. Під фоновим ІІІ (ambient invisible intelligence) мається на увазі широке поширення невеликих і недорогих міток і сенсорів, що дозволяють відстежувати розташування та статус різних об'єктів або навколишнього середовища. Ця інформація потім відправляється в хмару для аналізу та зберігання. Така технологія вбудовуватиметься у предмети повсякденного побуту, часто без відома користувача.

Прогноз Gartner: у період до 2028 року ранні зразки фонового ІІІ будуть застосовуватися для недорогого відстеження та виявлення предметів, що дозволить вирішувати нагальні проблеми бізнесу у сфері зниження витрат та підвищення продуктивності.

«На виробництві верстати та деталі зможуть взаємодіяти з системою управління, надсилаючи інформацію про режим роботи, запити на техобслуговування чи поповнення запасів. Це допоможе оптимізувати ланцюги поставок, скоротить прости обладнання та автоматизує замовлення сировини та матеріалів.» – Нік Джонс (Nick Jones), Gartner.

6. Під енергоефективними обчисленнями (energy-efficient computing) розуміється розробка та експлуатація комп'ютерів, центрів обробки даних та інших цифрових систем з метою мінімізації енергоспоживання та вуглецевого сліду.

Прогноз Gartner: вуглецевий слід – пріоритет для більшості ІТ-організацій.

«Уявіть собі розумний офіс, де енергоефективні обчислення інтегровані у кожен аспект повсякденних операцій. Сенсори на базі інтернету речей у реальному часі відстежують заповнюваність приміщень, регулюють освітлення, опалення та вентиляцію, що веде до істотного скорочення витрат та вуглецевого сліду.» – Нік Джонс (Nick Jones), Gartner.

7. Гібридні обчислення (hybrid computing) – це комбінування різних обчислювальних технологій, таких як центральні процесори (CPU), графічні процесори (GPU), прикордонні пристрої, спеціалізовані мікросхеми (ASIC), нейроморфні та квантові комп'ютери, фотоніка, для вирішення складних обчислювальних завдань. Гібридне середовище дозволяє скористатися перевагами кожної технології.

«Компанія може розмістити свої ключові програми на локальних серверах з міркувань безпеки, але при цьому використовувати хмарні технології для обробки великих даних, ШІ або резервного зберігання. Такий гібридний підхід дозволяє ефективно масштабувати діяльність, знижувати витрати та зберігати гнучкість.» – Соеб Барот (Soyeb Barot), Gartner.

8. Просторові обчислення (spacial computing) доповнюють фізичний світ цифровим контентом, дозволяючи користувачам взаємодіяти з ним – реалістично, інтуїтивно-зрозуміло, з ефектом занурення.

Прогноз Gartner: до 2028 року 20% людей хоча б раз на тиждень звертатимуться до контенту з перманентною геопозиційною прив'язкою порівняно з менш як 1% таких людей у 2023 році.

«За допомогою просторових обчислень компанії можуть створювати цифрові 3D-двійники фізичних активів. Це створює ефект занурення, що

допомагає контролювати та оптимізувати роботу, прогнозувати потребу в техобслуговуванні та тестувати обладнання, не перериваючи роботу.» – Марті Резнік (Marty Resnick), Gartner.

9. Багатофункціональні роботи (polyfunctional robots) – це гнучкі машини, здатні виконувати різноманітні операції, наслідуючи приклад або вказівку людини.

Прогноз Gartner: до 2030 року щодня взаємодіяти з розумними роботами будуть 80% людей, порівняно з менш ніж 10% сьогодні.

«Роботи, які працюють разом із людьми, мають бути пристосовані до середовища, спроектованого під людину. Це не обов'язково означає, що вони повинні бути людиноподібними – найбільш успішні мультифункціональні роботи не повторюватимуть форму тіла людини, а її вдосконалюватимуть.» - Білл Рей (Bill Ray), Gartner.

10. Нейроудосконалення (neurlogical enhancements) – це технології поліпшення когнітивних здібностей людини за рахунок зчитування та розшифрування сигналів мозкової активності або навіть надсилання сигналів у мозок.

Прогноз Gartner: нейроудосконалення, такі як двосторонній інтерфейс мозок-машина, що тільки з'явилися в 2024 році, до 2030 року охоплять 60% ІТ-фахівців.

1.2 Аналіз літератури

В роботі [3] висловлена основна ідея щодо створення вільних від логіки векторних обчислень, використовуючи лише транзакції читання-запису на адресній пам'яті. Стратегічною метою є створення детермінованих векторно-квантових обчислень з використанням фотонів для транзакцій читання-запису на стабільних елементах субатомної пам'яті. Основним завданням є реалізація нових моделей та методів векторних обчислень, заснованих на примітивних транзакціях читання-запису в технології

векторного гнучкого інтерпретативного моделювання та симуляції помилок, де дані використовуються як адреси для обробки самих даних. Суть векторних обчислень полягає в транзакціях читання-запису на векторних структурах даних в адресній пам'яті. Векторні обчислення – це обчислювальний процес, заснований на елементарних транзакціях читання-запису над комірками двійкових векторів, які зберігаються в адресній пам'яті та утворюють функціональність, де вхідними даними для обробки є адреси цих комірок. Визначено переваги векторної універсальної моделі для компактного опису впорядкованих процесів, явищ, функцій та структур з метою їх паралельного аналізу. Аналітичні вирази логіки, які потребують алгоритмічно складних калькуляторів, замінюються вихідними векторами стану елементів та цифровими схемами, орієнтованими на паралелізм регістрових логічних процедур на регулярних структурах даних. Запропоновано векторно-дедуктивний метод синтезу формул для поширення вхідних списків (даних) несправностей, який має квадратичну обчислювальну складність регістрових операцій. Синтезовано нову матрицю дедуктивних векторів, яка характеризується такими властивостями: компактність, паралельна обробка даних на основі однієї транзакції читання-запису в пам'яті, виключення традиційної логіки з процедур моделювання несправностей, повна автоматизація процесу її синтезу та орієнтація на технологічне вирішення всіх задач технічної діагностики. У роботі здійснено перехід до векторної логіки в організації обчислень та виключення традиційної логіки, представлені у вигляді таблиць та аналітичних виразів. Використання транзакцій читання-запису в пам'яті за відсутності системи команд орієнтує нові векторно-логічні обчислення в бік детермінованих квантових архітектур, заснованих на стабільних субатомних частинках пам'яті.

В [4] основним завданням є реалізація нових простих та надійних моделей і методів векторних обчислень на основі примітивних транзакцій читання-запису в технології векторного гнучкого інтерпретативного

модельовання несправностей цифрових систем. Векторно-логічні обчислення – це обчислювальний процес, заснований на транзакціях читання-запису над бітами двійкового вектора функціональності, де вхідними даними є адреси бітів. Запропоновано векторний метод синтезу дедуктивних матриць для транспортування вхідних списків несправностей, який має квадратичну обчислювальну складність. Метод є розвитком алгоритму дедуктивного синтезу векторів на основі таблиці істинності. Дедуктивна матриця призначена для синтезу та верифікації тестів з використанням паралельного модельовання несправностей, як адрес, на основі транзакції читання-запису дедуктивних комірок вектора в пам'яті.

В роботі [5] запропоновано новий кубітний метод для генерації тестів дискретних функцій в компонентах SoC. Цей метод використовує булеві похідні, представлені у векторній формі, для опису поведінки логічних елементів у вигляді Q-покриття. Підкреслено важливість математичних та технологічних зв'язків у структурі даних, що дозволяє розробляти ефективні алгоритми та методи обробки даних для підвищення продуктивності тестування. Введено векторну модель булевих похідних, яка застосовується для створення дедуктивних матриць у кубітному методі модельовання несправностей та оцінки якості тестових послідовностей. Розроблено деревоподібний процесор ATPG, що представляє собою двійковий деревоподібний граф хог-елементів, для паралельної обробки частин кубітного покриття та структур даних логіки SoC при обчисленні булевих похідних кубіту. Запропоновані структури даних та методи реалізовані у програмному забезпеченні, яке зосереджено на паралельному тестуванні логічних функцій цифрових систем на кристалах з використанням кубітного покриття.

В [6] описано підхід до тестування та діагностики цифрових схем, зокрема систем на кристалі (SoC), що базується на використанні кубітних моделей логічних елементів XOR та NXOR. Запропоновано методи представлення цифрових пристроїв у вигляді кубітних моделей, що дозволяє

моделювати їхню поведінку та виявляти несправності. Для підвищення ефективності процесу розроблено паралельні алгоритми мінімізації логічних функцій, діагностики несправностей SoC та оптимізації покриття тестами. В основі цих алгоритмів лежить використання унітарного кодування для представлення даних у кубітній формі. Представлено також архітектуру сервісів, призначених для проектування, тестування та верифікації цифрових пристроїв, використовуючи запропоновані кубітні моделі. Одним з сервісів є інструмент для моделювання безпомилкової роботи схем, який враховує кубітне покриття функціональних примітивів. Слід зазначити, що всі розроблені моделі, структури даних та методи реалізовані та протестовані на класичних комп'ютерах, використовуючи унітарне кодування для представлення двійкових станів кубітів.

Монографія [7] представляє новаторський погляд на кіберкультуру, охоплюючи мікро-, макро-, космологічні та віртуальні обчислення. Вона пропонує унікальний підхід до моделювання та управління складними системами, використовуючи перетворення таблиць істинності та матриць суміжності в вектор кубітів для оптимізації обчислень, керованих пам'яттю. Автори розробили метрику для вимірювання процесів у кіберпросторі та архітектуру логічних асоціативних обчислень для ефективного аналізу великих даних та прийняття обґрунтованих рішень. Книга також представляє інноваційні методи проектування, тестування та діагностики цифрових систем, засновані на векторі покриття кубітів, та технологію діагностики HDL-моделей SoC за допомогою графа тестових тверджень. Наведено практичні приклади застосування кіберфізичних систем для моніторингу та управління соціальними об'єктами та транспортом.

В [8] векторне моделювання несправностей використовується як технологія для оцінки якості тестів складних IP-ядер, реалізованих у програмованих вентильних матрицях (FPGA), спеціалізованих інтегральних схемах (ASIC). Аналітичні вирази логіки, що потребують алгоритмічно складних обчислень, замінюються векторами вихідних станів елементів та

цифрових схем. Синтезовано нову матрицю дедуктивних векторів, яка характеризується такими властивостями: компактність, паралельна обробка даних на основі однієї транзакції читання-запису в пам'яті, виключення традиційної логіки з процедур моделювання несправностей, повна автоматизація процесу її синтезу та орієнтація на технологічне вирішення багатьох задач технічної діагностики. Запропоновано нову структуру секвенсора для векторного дедуктивного моделювання несправностей, яка характеризується простотою реалізації на одному блоці пам'яті. Вона виключає будь-яку традиційну логіку, використовує транзакції читання-запису даних у пам'яті для формування вихідного вектора несправностей, використовує дані як адреси для обробки самих даних.

Основним завданням статті [9] є створення векторно-логічного обчислювального середовища, яке для моделювання несправностей використовує лише операції читання-запису (read-write) в адресній пам'яті, де кожна несправність розглядається як адреса. У цьому підході відсутня традиційна логіка. Такий підхід також не потребує команд процесора чи арифметико-логічного пристрою (АЛП) для організації обчислень, що робить його придатним для реалізації на кристалах SoC та FPGA. Запропоновано векторно-логічний метод синтезу дедуктивних матриць для поширення вхідних несправностей, який має квадратичну обчислювальну складність. Розроблено векторний дедуктивний метод моделювання несправностей, що базується на примітивних операціях читання-запису, для аналізу логічних схем. Для цього використовується набір вхідних тестів та логічний вектор функціональності. Цей метод є розвитком алгоритму синтезу дедуктивних векторів на основі таблиці істинності. Дедуктивна матриця використовується для синтезу та верифікації тестів шляхом паралельного моделювання комбінацій несправностей, представлених як адреси, за допомогою операцій читання-запису над бітами дедуктивних векторів, що зберігаються в пам'яті. Запропоновано векторний метод синтезу дедуктивних матриць для транспортування вхідних векторів несправностей на вихід елемента.

Розроблено структури даних для паралельного моделювання несправностей цифрових схем, використовуючи примітивні операції читання-запису в матричній пам'яті, де комбінації несправностей представлені стовпцями-адресами. Запропоновано секвенсор із п'яти блоків, що складають векторно-логічний обчислювальний пристрій, який використовується для дедуктивного моделювання несправностей на основі операцій читання-запису. Проведено верифікацію розроблених моделей та методів на тестових прикладах. Вперше запропоновано векторно-логічний метод синтезу матриці дедуктивних векторів для паралельного моделювання комбінацій вхідних несправностей, представлених як адреси. Вперше запропоновано автомат векторно-дедуктивного моделювання несправностей, представлених як адреси, на основі операцій читання-запису, який призначений для реалізації в FPGA LUT. Також запропоновано вбудований online симулятор SoC, як ядро для моделювання несправностей цифрових систем RTL-рівня. Продемонстровано технологічні переваги векторно-логічного синтезу дедуктивних матриць на численних прикладах традиційної та RTL-логіки, що підкреслює технологічність векторів у порівнянні з аналітичними дедуктивними формулами для побудови симуляторів. Матриця дедуктивних векторів, як сукупність вектор-стовпців булевих похідних, використовується для побудови мінімальних тестів для логічних елементів. Розроблено рекурсивну формулу синтезу матриці перестановки координат у логічному векторі активності, що дозволяє значно спростити отримання дедуктивної матриці для моделювання несправностей, представлених як адреси. Практична користь in-memory симулятора полягає в значному підвищенні продуктивності моделювання несправностей цифрових блоків SoC, що дозволяє досягти швидкості порядку сотень наносекунд. Надано оцінки обчислювальних витрат, пов'язаних з використаними алгоритмами.

В [10] Основна ідея полягає у створенні векторного моделювання без логіки, що базується лише на транзакціях читання-запису адресної пам'яті. Вектор моделювання із застряглими несправностями використовується як

технологія оцінки якості тестів для складних IP-ядер, реалізованих Field Programmable Gate Array (FPGA), Application-Specific Integrated Circuit (ASIC). Основне завдання полягає у реалізації нових простих та надійних моделей та методів векторних обчислень на основі примітивних транзакцій читання-запису у технології векторного гнучкого інтерпретативного моделювання несправностей. Векторні обчислення є обчислювальним процесом, заснованим на транзакціях читання-запису бітів двійкового вектора функціональності, де вхідними даними є адреси бітів. Запропоновано векторно-дедуктивний метод синтезу векторів для розповсюдження списків несправностей, що має квадратичну обчислювальну складність. Аналітичні висловлювання логіки, що вимагають алгоритмічно складних обчислень, замінюються векторами вихідних станів елементів та цифрових схем. Синтезовано нову матрицю дедуктивних векторів, яка характеризується такими властивостями: компактність, паралельна обробка даних на основі однієї транзакції читання-запису в пам'яті, виключення традиційної логіки з процедур моделювання несправностей, повна автоматизація процесу її синтезу та орієнтація на технологічне вирішення багатьох завдань технічної діагностики. Запропоновано нову структуру секвенсора для дедуктивного моделювання несправностей, яка характеризується простотою реалізації на одному блоці пам'яті. Вона виключає будь-яку традиційну логіку, використовує транзакції читання-запису даних у пам'яті для формування вихідного вектора несправностей, використовує дані як адреси для обробки самих даних.

У роботі [11] розглянуто розподілені генетичні алгоритми для задачі генерації тестів. Досліджено різні форми паралелізації генетичних алгоритмів для генерації тестів.

Логічне диференціальне числення – це галузь математики, аналогічна класичному диференціальному численню. Одним з його інструментів є логічні похідні, які дозволяють досліджувати, як змінюється значення дискретної функції при зміні значення змінної. Метод обчислення цих

похідних залежить від представлення функції. У статті основна увага приділяється функціям, представленим багатозначними діаграмами рішень (БДР). Спочатку представлено обчислення похідних за допомогою стандартних алгоритмів для маніпулювання БДР. Згодом представлено новий універсальний алгоритм для ефективнішого обчислення різних типів похідних. У роботі [12] представлено експериментальне порівняння нового алгоритму зі стандартним підходом. Результати порівняння показують, що новий алгоритм значно швидший.

Стаття [13] присвячена булевим та векторним бент-функціям та їхнім дуальним функціям. Мета – розширити корпус таких функцій шляхом розробки нових, що охоплюють багато попередніх конструкцій бент-функцій. З цією метою автори пропонуєть кілька нових нескінченних сімейств бент-функцій, включаючи ідемпотентні бент-функції будь-якого алгебраїчного ступеня, бент-функції в одновимірній формі сліду та самодвійні бент-функції. Ці бент-функції мають великий теоретичний та практичний інтерес завдяки їхнім особливим структурам та зв'язку з самодвійними кодами. Зокрема, багато відомих бент-функцій є окремими випадками запропонованих авторами бент-функцій. Крім того, автори поширюють свої результати на векторні бент-функції та отримують три нові нескінченні класи векторних бент-функцій будь-якого можливого ступеня, визначаючи явні дуальні функції трьох класів відомих бент-функцій.

1.3 Висновки та постановка завдання

Наведено тенденції розвитку сучасних технологій, стратегічні технологічні тренди, які визначають майбутнє, стимулюючи інновації, згідно до аналітичного огляду консалтингової компанії Gartner. Робиться висновок, що актуальними для комп'ютерної інженерії є пункти 6 – 8 щодо енергоефективних обчислень (розробка та експлуатація комп'ютерів), гібридних обчислень як комбінування різних обчислювальних технологій –

центральні процесори (CPU), графічні процесори (GPU), спеціалізовані мікросхеми (ASIC), квантові комп'ютери, а також просторові обчислення. У зв'язку з цим питання діагностики обчислювальних пристроїв є затребуваними.

Метою бакалаврської роботи є підвищення ефективності виявлення несправностей при проектуванні цифрових систем шляхом синтезу схем діагностики несправності вхідної змінної на основі векторно-логічних обчислювальних механізмів.

Задачі: огляд стану технологій; аналіз публікацій, дослідження моделей, методів, технологій векторно-логічного комп'ютингу; синтез схем діагностики вхідної змінної на основі теорії векторно-логічного комп'ютингу.

2 МАТЕМАТИЧНИЙ АПАРАТ ТА ТЕХНОЛОГІЧНІ РІШЕННЯ ВЕКТОРНО-ЛОГІЧНОГО КОМП'ЮТИНГУ

Розглядаються основні поняття та визначення, актуальні моделі, метрики комп'ютингу, розумні структури, математичний апарат векторно-логічного комп'ютингу для подальшого синтезу схем діагностики несправності вхідної змінної на основі векторно-логічних обчислювальних механізмів [4-10].

2.1 Визначення комп'ютингу

Метрика – спосіб вимірювання відстані (подібності-відмінності) між процесами або явищами (об'єктами) у загальному просторі заданих параметрів.

Якість – сукупність метричних властивостей процесу чи явища, визначальних його придатність задовольняти заданим потребам відповідно до призначенням.

Явище – інтегральне чуттєво-логічне сприйняття людиною фрагментів навколишнього світу. Метрика явища – відстань між компонентами у просторі.

Процес – упорядкована у часі послідовність дій (для досягнення мети). Метрика процесу – відстань між компонентами у часі.

Загальні властивості процесів чи явищ полягають у відновлюваній сукупності елементарних компонентів-примітивів у часі та просторі, що становлять універсум. Тому будь-який процес чи явище можна легко уявити логічною функціональністю на універсумі примітивів.

Синтез логічної функціональності процесу чи явища: унітарне кодування n -паттернів на універсумі примітивів формує двійкові вектори таблиці істинності, яка своїми ненульовими стовпцями-адресами еквівалентних примітивів формує поодинокі значення логічного вектора.

Модель (структури даних) – сукупність відносин між компонентами, із заданою адекватністю описує метричні властивості процесу чи явища.

Алгоритм – часова послідовність дій моделі для досягнення результату.

Механізм – це сполучені відносини між надмірністю структур даних і обчислювальною складністю алгоритмів їх обробки, коли збільшення одного з них призводить до зменшення іншого і навпаки.

Метод – сукупність дій у просторі та часі на моделях для досягнення мети.

Підхід – сукупність методів та моделей для дослідження та перетворення процесів та явищ на основі інтегруючої ідеї.

Система (комп'ютинг) – сукупність відносин між механізмами управління-виконання та довіллям з вираженими функціями моніторингу та актуації задля досягнення поставлених цілей.

Векторно-логічний комп'ютинг (vector logic computing) – економічні механізми інтелектуального in-memory комп'ютингу з використанням read-write транзакцій для вирішення практичних завдань аналізу та управління фізичними, соціальними, бізнес-процесами на основі моніторингу. Для будь-якого завдання можна побудувати таку надмірну модель, яка обнуляє алгоритм її аналізу для отримання рішення [8, 10].

Векторно-логічний комп'ютинг призначений для істотного зменшення часових та енергетичних витрат при управлінні процесами та явищами у фізичному, соціальному та цифровому світі. Використовуються механізми векторно-логічного in-memory комп'ютингу, що істотно спрощує алгоритми за рахунок експоненційної надмірності структур даних [9].

Інтелектуальний комп'ютинг – як гармонійні відносини у будь-якому обчислювальному середовищі (cloud, fog, edge) між алгоритмом та моделлю для отримання рішення за метрикою економії часу та енергії шляхом використання будь-яких архітектур: AI, фон Неймана, квантового або in-memory комп'ютингу [9].

Промпт-комп'ютинг призначений для верифікації цифрових функціональностей шляхом побудови карти та тестування за логічним вектором. Використовуються моделі та алгоритми моделювання несправностей цифрових логічних схем з урахуванням дедуктивних векторів, одержуваних з логічних векторів елементів схеми.

2.2 Design and Test комп'ютинг

Використовується карта тестування, як продукт modeling and simulation. Карта синтезується за логічним вектором функціональності і є смарт-моделлю для design and test промпт-інжинірингу.

Карта тестування – це механізм миттєвого моделювання всіх несправностей на вичерпному тесті в результаті створення розумних структур даних. Адреса – це утворюючий фактор in-memo комп'ютингу, модель комбінацій несправностей, тестів та «прихований» високий механізм обробки даних. Адреса таблиці істинності є ключовою моделлю для перенесення в лінійний простір складності алгоритмів для вирішення комбінаторних проблем. Адреса – це не лише атрибут пам'яті та логічного вектора, але й таблиця істинності для запису тесту та формування повної комбінаторної моделі несправностей. У загальному випадку – intelligent computing – це синтез моделі на вхід-вихідних наборах для подальшого моделювання будь-яких вхідних наборів.

Intelligent computing is modeling for simulation. Modeling – це синтез моделі (логічного вектора) на вхідних наборах. Simulation – це визначення стану виходу моделі за вхідним набором та синтезованою моделлю (логічним вектором).

2.3 Розумні структури даних

Розумні структури даних – це структури даних, з'єднані єдиною метрикою логічного векторного простору для вирішення задачі без

програмування. Для моделювання несправностей – це явні структури даних (вектори, таблиці, матриці), пов'язані єдиною (уніфікованою) метрикою логічного векторного простору 2^n .

Мета – підвищення продуктивності та якості моделювання несправностей логічних схем шляхом суперпозиції таблиці істинності одиночних та кратних константних несправностей та матриці дедуктивних векторів.

Завдання: 1) розробка моделі вхідних несправностей визначення якості тесту; 2) суперпозиція таблиці істинності матриці дедуктивних векторів для аналізу якості тесту; 3) алгоритм спільного аналізу матриці дедуктивних векторів та таблиці істинності вхідних одиночних та кратних константних несправностей.

Вихідні дані: 1) логічний вектор функціональності Q та двійковий тестовий набір x -test; 2) похідні – матриця дедуктивних векторів D та таблиця істинності X одиночних та кратних вхідних константних несправностей.

На рис. 2.1 представлено суперпозицію однакових за розмірами матриць дедуктивних векторів та таблиці істинності всіх комбінацій константних несправностей на входах логічного елемента, зображеного ліворуч у двох формах: 1) реалізація термів ДНФ; 2) елемент з урахуванням логічного вектора.

Стовпці таблиці істинності X одиничними координатами формують вичерпні комбінації одиночних та кратних константних дефектів, які є похідними від двійкового вхідного тестового набору.

Символ $X_{ij}=1$ означає одиночну несправність j^{X_i} . Карта несправностей ідентифікована декількома 1-координатами в стовпці матриці X .

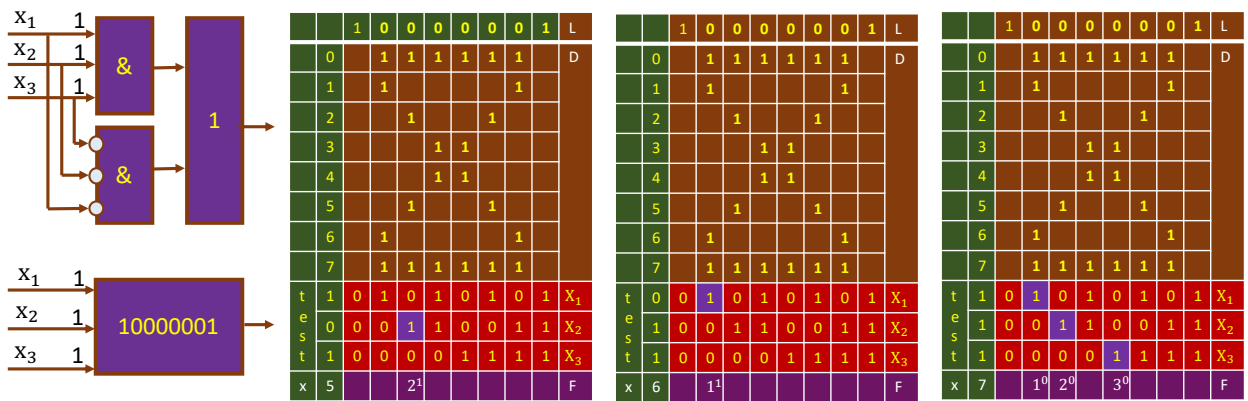


Рисунок 2.1 – Суперпозиція матриць дедуктивних векторів

Наприклад, у фіолетових клітинах одиницями зазначені несправності, які перевіряються на тестових наборах $x=101(2^1), 011(1^1), 111(1^0, 2^0, 3^0)$.

Алгоритм формування координат вектора дефектів (рис. 2.2), що перевіряються, містить один оператор логічного множення X-матриці на вектор D_x : $L=X \wedge D_x$, при цьому дедуктивний вектор D_x ($x=5, 6, 7$) розглядається як вектор стану виходів таблиці істинності 101, 011, 111. Результат множення своїми одиничними координатами формує сукупність несправностей, що перевіряються, які ідентифікуються в останніх рядках таблиць за загальноприйнятим позначенням.

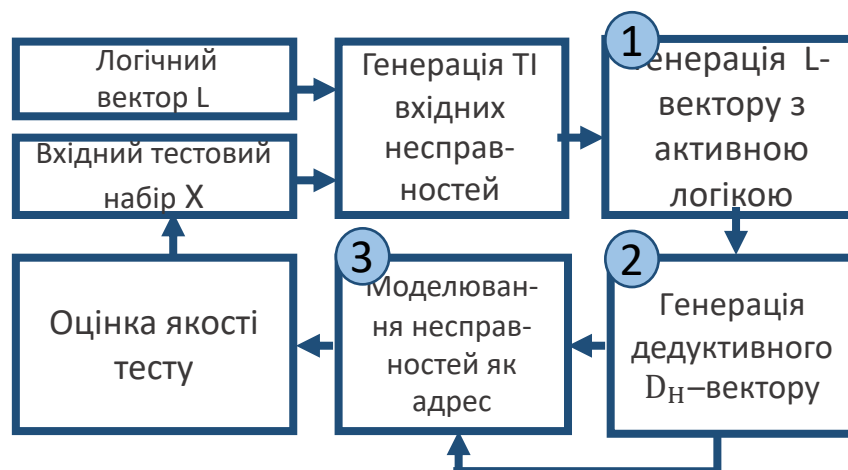


Рисунок 2.2 – Алгоритм формування координат векторів дефектів

Таким чином, розумні структури даних представлені для моделювання несправностей як адрес у цифрових схемах. Компонентами структури є: матриця моделювання, опис зв'язків схеми, логічні та дедуктивні вектори. У такому вигляді використовують структури даних для формування результатів моделювання несправностей на одному вхідному наборі.

2.4 Механізм адресного заповнення Fault Simulation Matrix

Метрика новизни FAAS (Faults-as-address circuit simulation) механізму – це адресне заповнення Fault Simulation Matrix за допомогою дедуктивних векторів, які моделюють комбінації несправностей, як адрес.

Відношення

$$T \oplus L = D \quad (2.1)$$

між тестовим набором T та таблицею істинності L елементу формує дедуктивний вектор для моделювання несправностей, як адрес таблиці істинності або координат логічного вектора.

З іншого боку, адреси використовуються в матриці моделювання для позначення n -комбінацій вхідних несправностей, які перевіряються на виході елемента. Фактично, на дедуктивному векторі елемента обробляються стовпці, як адреси n -рядків матриці моделювання для того, щоб згенерувати один вихідний рядок елемента. Жодного транспортування вхідних несправностей на вихід елемента не відбувається. Виконується лише заповнення одиницями невхідних рядків матриці моделювання схеми. Матриця моделювання схеми спочатку заповнюється одиницями по головній діагоналі. Перевірені на тестовому наборі несправності лінії схеми визначаються інверсними значеннями справного стану ліній, які мають одиничні значення в рядку матриці, що відповідає виходу останнього елемента схеми. Фактично, дедуктивний вектор виходить хог-відношень між тестовим набором та логічним вектором за три табличні операції.

Перевага FAAS-механізму – це передбачувана складність алгоритму та витрати пам'яті для зберігання структур даних при моделюванні тестового набору, яка визначається за формулою:

$$Q = N^2 + \sum_{i=1}^n (Y_i + D_i), \quad (2.2)$$

N , n – кількість ліній та елементів в схемі, Y , D – множина логічних и дедуктивних векторів.

2.5 Таблиця істинності для моделювання несправностей

Таблиця істинності використовується для моделювання вхідних несправностей будь-якого логічного елемента. Формула моделювання: за вхідним двійковим словом, що використовується як адреса комірки логічного вектора, знаходиться стан виходу елемента, який конкатенується з вхідним словом для формування вектора справного стану, призначеного для хог-взаємодії з усіма стовпцями таблиці істинності з метою формування невпорядкованої за адресами таблиці та бітів вектора активності за адресами у відповідних стовпцях таблиці, після цього отриманий дедуктивний вектор своїми одиничними значеннями активує одиничні координати у відповідних стовпцях таблиці істинності несправностей, знаки яких визначаються інверсними станами вхідних змінних.

Визначення переліку вхідних несправностей, що підлягають перевірці логічним елементом, шляхом генерації дедуктивного вектора для двійкової вхідної множини без перекодувальної матриці. Вихідні дані: вхідний двійковий набір, довжина n : $T=11$ та логічний вектор $Y=0111$, довжиною 2^n , що формує 2^n вхідних наборів таблиці істинності X , на прикладі змінних $x_1 x_2$.

Вихідні дані: таблиця істинності перевірюваних несправностей, розмірності $n \times 2^n$. Розмір пам'яті для зберігання структур даних визначається виразом:

$$M = n \times 2^n + 2^n + n + 1, \quad (2.3)$$

Де n – кількість змінних логічного елемента, який утворює компактні та інтелектуальні структури даних: таблицю істинності, логічний (дедуктивний) вектор, вхідне двійкове слово $x=11$ та вихідні стани $Y=1$.

1. Моделювання справної поведінки елемента – значення виводу Y обчислюється при подачі двійкового вхідного набору T , що розглядається як адреса біта логічного вектора $Y=Q_T$.

2. Векторно-матрична операція $L=T$ для отримання таблиці активності L , де стовпці, що відповідають значенням змінних x_1x_2 , розглядаються як двійкові адреси бітів дедуктивного D -вектора.

3. Упорядкування координат L -вектора за вхідними двійковими адресами x_1x_2 таблиці істинності для отримання дедуктивного D вектора

$$D_{x_i} = L_i. \quad (2.4)$$

Ця операція потрібна для приведення таблиці істинності до прийнятого стандарту впорядкованих двійково-десяткових адрес на основі вхідних змінних, що необхідно для обробки елементів в цифровій структурі. Обчислювальна складність побудови дедуктивного вектора:

$$2^n \times (n + 1). \quad (2.5)$$

4. Генерація перевірюваних вхідних несправностей логічного елемента за оберненими значеннями вхідних змінних сигналів x_1x_2 на одиничних координатах стовпців таблиці істинності, що покриваються (активізованих) одиничними значеннями координат дедуктивного вектора:

$$F_{ij} = \bar{T}_i \leftarrow F_{ij} \wedge D_i = 1. \quad (2.6)$$

Обчислювальна складність алгоритма визначається наступною формулою:

$$Q = 1 + 2^n \times (n + 1) + 2^{n+1} + 2^n \times n \quad (2.7)$$

виконання read-write транзакцій.

2.6 Апарат булевих похідних для синтезу тестів функціональності за логічним вектором

Булева похідна першого порядку функції $f(x_1, x_2, \dots, x_n)$ по змінній x_i показує, коли зміна значення x_i призводить до зміни значення самої функції f . При цьому, отримана булева похідна не залежить від змінної x_i , по якій вона обчислювалася:

$$\frac{df}{dx_i} = f(x_1, x_{i-1}, 1, x_{i+1}, x_n) \oplus f(x_1, x_{i-1}, 0, x_{i+1}, x_n). \quad (2.8)$$

Алгоритм обчислення булевої похідної диз'юнктивної нормальної форми має кубічну часову складність відносно кількості змінних n : $Q = n^3$ [11-13].

Для схеми (рис. 2.3), яка відповідає виразу

$$f = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3, \quad (2.9)$$

булеві похідні першого порядку за всіма трьома змінними визначені відповідно як:

$$\frac{df}{dx_1} = x_2 x_3; \quad (2.10)$$

аналітичних формул використовувати логічні вектори, що описують функції та схеми, обчислення булевої похідної стає значно простішим.

Карта тестування функціональності будується за допомогою дедуктивної матриці. Ця матриця містить всі можливі комбінації булевих похідних різних порядків, обчислених для кожної окремої координати таблиці істинності несправності.

Для отримання аналітичної форми булевих похідних, необхідно мінімізувати вектори похідних (стовпці дедуктивної матриці). Рядки матриці представляють собою дедуктивні вектори, які описують поширення вхідних несправностей на вихідні дані схеми.

Булева похідна – це інструмент, що дозволяє виявити, які частини логічної функції залишаються незмінними, коли одна або декілька її вхідних змінних змінюються. Іншими словами, вона показує, які умови повинні бути виконані, щоб зміна вхідних даних не вплинула на вихід.

Булеву похідну можна розглядати як міру подібності логічної функції, оскільки вона виявляє умови, за яких функція поводить себе передбачувано, незалежно від зміни певних вхідних даних. У контексті тестування, одиничні значення в рядку булевої похідної вказують на те, які несправності можуть бути виявлені на виході логічного елемента. Одиничні значення у стовпці, навпаки, показують, які вхідні умовинеобхідні для того, щоб несправність, що існує на вході, була "передана" на вихід.

Таким чином, рядок булевої похідної можна інтерпретувати як дедуктивний вектор, що показує, які несправності виявляються, а стовпець – як власне булеву похідну, що визначає умови для передачі несправностей (рис. 2.4).

X_1	X_2	X_3	D (L=01111111)							
		X_1	0	1	0	1	0	1	0	1
		X_2	0	0	1	1	0	0	1	1
		X_3	0	0	0	0	1	1	1	1
0	0	0		1	1	1	1	1	1	1
1	0	0		1						
0	1	0			1					
1	1	0				1				
0	0	1					1			
1	0	1						1		
0	1	1							1	
1	1	1								1
Derivatives →				X'_1	X'_2	X'_{12}	X'_3	X'_{13}	X'_{23}	X'_{123}

Рисунок 2.4 – Синтез булевих похідних за логічним вектором 01111111

Використання значень першого стовпця як вхідних даних для алгоритму мінімізації булевої функції трьох змінних призводить до наступного спрощеного виразу:

$$Q'(x_1) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3 = (\bar{x}_1 \vee x_1) \bar{x}_2 \bar{x}_3 = \bar{x}_2 \bar{x}_3. \quad (2.13)$$

Інтерпретація результату – активізація першого входу можлива лише у випадку, якщо на двох інших входах буде встановлено два нулі.

2.7 Висновки до розділу 2

Виконано аналіз математичного апарату та технологій векторно-логічного комп'ютингу для подальшого синтезу схем діагностики несправності вхідної змінної.

3 СИНТЕЗ СХЕМ ДІАГНОСТИКИ НЕСПРАВНОСТЕЙ ВХІДНОЇ ЗМІННОЇ

Виконується синтез схем діагностики вхідної змінної на основі теорії векторно-логічного комп'ютингу. Розглядається синтез схем діагностування несправностей вхідних змінних з використанням всіх булевих похідних логічної функції. Наводяться приклади синтезу схем діагностування одиночних несправностей вхідних сигналів для логічної функції з використанням дедуктивної матриці.

3.1 Використання булевих похідних для тестування цифрових схем

Булева похідна є інструментом у тестуванні цифрових схем, оскільки дозволяє:

1) ідентифікувати умови активації виходу – визначати, які зміни на входах (одного або декількох) призводять до зміни стану виходу елемента. Це дозволяє простежити логічний шлях проходження сигналу;

2) створювати ефективні тести – синтезувати мінімальний набір тестів, що перевіряють всі можливі шляхи до виходу елемента. Для кожного входу, використовуючи булеву похідну, можна підібрати пари вхідних сигналів (0,1), які активують відповідний логічний шлях і виявляють можливі константні несправності на шляху від входу до виходу;

3) розробляти схеми діагностики – створювати спеціалізовані блоки або схеми, які діагностують несправності на зовнішніх входах під час нормальної роботи цифрової схеми. Такі системи особливо важливі в критичних застосуваннях, таких як управління ядерними реакторами або транспортними засобами, де надійність має першорядне значення;

4) оптимізувати проектування – визначати, які вхідні змінні є важливими для функціонування схеми, а які ні. Це знання є критичним при проектуванні схем управління різними процесами та явищами, дозволяючи спростити схему та підвищити її ефективність.

3.2 Визначення несуттєвості вхідної змінної за булевою похідною

Розглядається синтез схем діагностування несправностей вхідних змінних на основі всіх булевих похідних логічної функції.

Булеві похідні логічної функції, представлені у вигляді стовпців дедуктивної матриці, показують, за яких умов зміна значення однієї або декількох вхідних змінних гарантовано призведе до зміни вихідного значення логічного елемента або схеми.

Стовпець дедуктивної матриці, що містить хоча б одну одиницю, є булевою похідною. Якщо ж стовпець складається лише з нулів, то відповідна вхідна змінна (або їх комбінація) не впливає на вихід логічної функції (рис. 3.1).

	0	0	1	1		H					D	00	01	10	11
0			1	1		0	1	2	3		00			1	1
0			1	1		1	0	3	2		01			1	1
1	1	1				2	3	0	1		10			1	1
1	1	1				3	2	1	0		11			1	1
Derivatives →													X' ₂	X' ₁	X' ₁₂

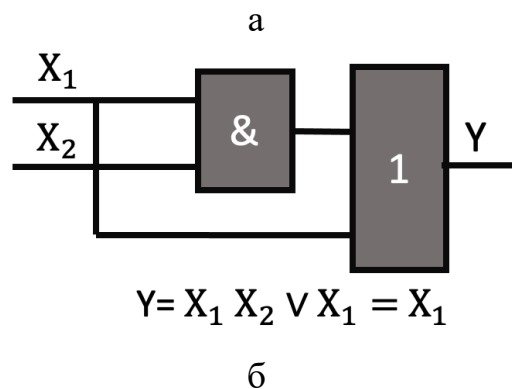


Рисунок 3.1 – Несуттєвість вхідної змінної, що визначена за булевою похідною

Приклад 3.1. Розглядається дедуктивна матриця для логічної функції для трьох вхідних змінних 11110000.

Підхід: 1) аналіз булевих похідних – використовуються лише стовпці дедуктивної матриці, що відповідають булевим похідним для кожної вхідної змінної. Комбінації цих похідних не враховуються; 2) створення схем для похідних – для кожного стовпця (тобто для кожної похідної) розробляється окрема схема, яка визначає умови, за яких ця похідна існує (активна); 3) забезпечення шляхів для несправностей – ці схеми гарантують наявність активізованих логічних шляхів між кожною вхідною змінною та виходом схеми. Результат: кожен логічний шлях, створений таким чином, передає будь-яку несправність на вході на вихід елемента. Це дозволяє виявити несправність вхідного сигналу під час роботи схеми.

На основі чотирьох стовпців дедуктивної матриці синтезовано чотири схеми транспортування несправностей на вихід схеми, що дозволяють виявляти такі несправності під час роботи пристрою (рис. 3.4).

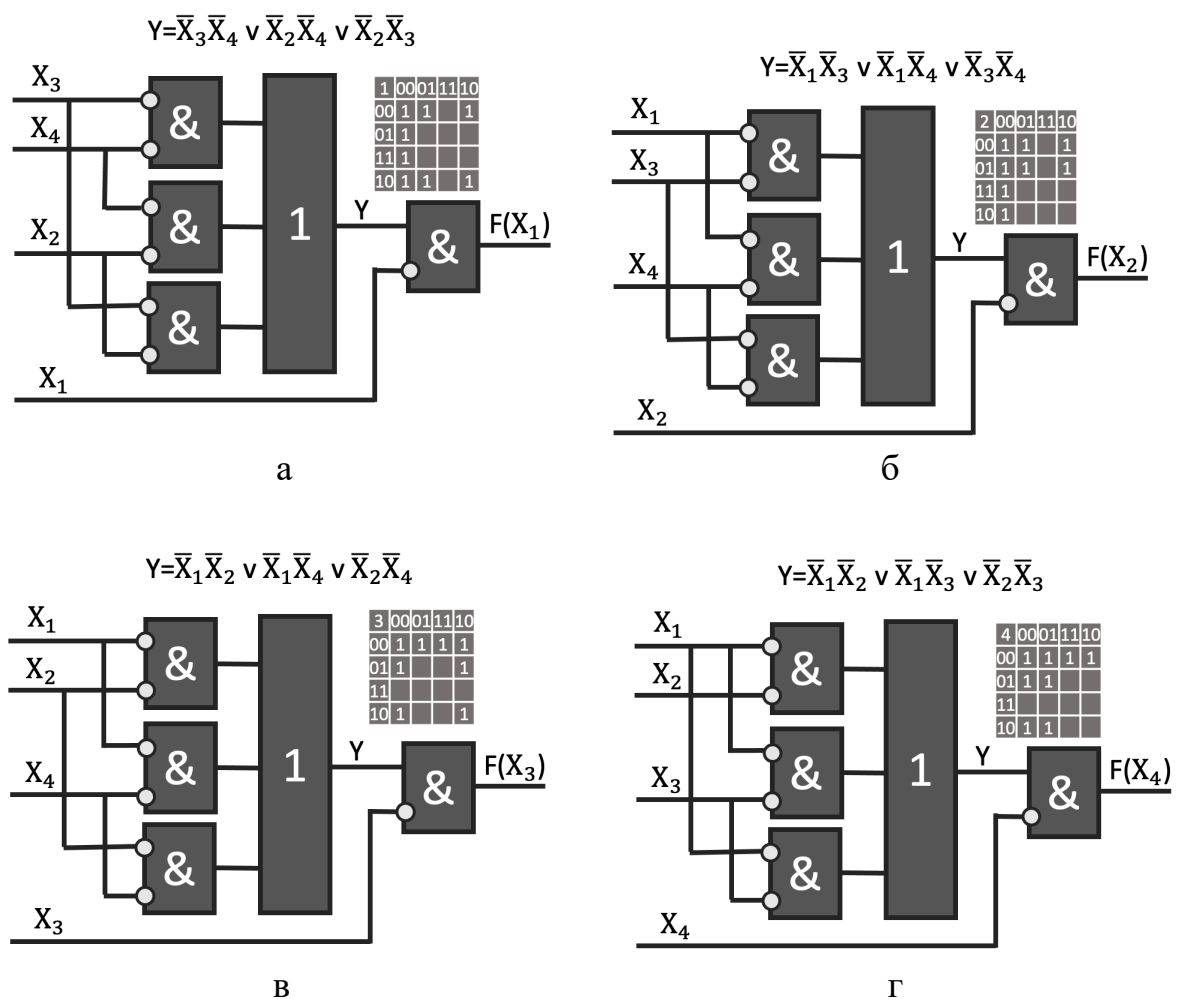


Рисунок 3.4 – Схеми діагностування вхідних несправностей

Однак такий класичний підхід до синтезу схем діагностування може бути замінений in-memory комп'ютингом, коли розглядаються чотири стовпці дедуктивної матриці, вміщені в пам'ять, для транспортування дефектів вхідних змінних. В цьому випадку схема діагностування може бути представлена двома наступними блоками (рис. 3.5).

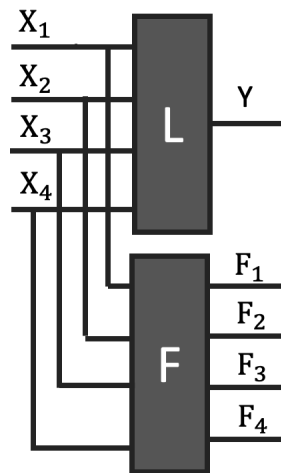


Рисунок 3.5 – Схема для чотирьох булевих похідних із дедуктивної матриці для діагностування

Дедуктивна матриця показує, як різні несправності впливають на результат роботи логічної схеми. Інформація, що міститься в її стовпцях, які описують вплив кожної вхідної змінної та їх комбінацій, може бути використана для синтезу схем, з метою виявлення несправності вхідних сигналів в режимі реального часу.

Становить інтерес методика виявлення несправності за дедуктивною матрицею, оскільки така матриця формує відношення між вичерпним тестом і всіма комбінаціями несправностей (рис. 3.6).

T↔F		0	1	2	3	4	5	6	7	F _U	F _∩
TT-Faults	x ₁		1		1		1		1	1	.
	x ₂			1	1			1	1	1	.
TT-test	x ₃					1	1	1	1	1	.
N	x ₁	x ₂	x ₃								
0	0	0	0							1	
1	1	0	0					1			
2	0	1	0				1				
3	1	1	0			1					
4	0	0	1			1					
5	1	0	1		1						
6	0	1	1		1						
7	1	1	1		1	1	1	1	1	1	

а

T↔F		0	1	2	3	4	5	6	7	F _U	F _∩
TT-Faults	x ₁		0		0		0		1	x	.
	x ₂			0	0			0	1	x	.
TT-test	x ₃					0	0	0	1	x	.
N	x ₁	x ₂	x ₃								
0	0	0	0							1	
1	1	0	0					1			
2	0	1	0				1				
3	1	1	0			1					
4	0	0	1			1					
5	1	0	1		1						
6	0	1	1		1						
7	1	1	1		1	1	1	1	1	1	

б

T↔F		0	1	2	3	4	5	6	7	F _U	F _∩
TT-Faults	x ₁		1		0		0		1	1	1
	x ₂			0	0			0	1	1	.
TT-test	x ₃					0	0	0	1	1	.
N	x ₁	x ₂	x ₃								
0	0	0	0							1	
1	1	0	0					1			
2	0	1	0				1				
3	1	1	0			1					
4	0	0	1			1					
5	1	0	1		1						
6	0	1	1		1						
7	1	1	1		1	1	1	1	1	1	

в

Рисунок 3.6 – Діагностування несправностей за дедуктивною матрицею

Процес виявлення та локалізації несправностей за допомогою дедуктивної матриці включає наступні кроки.

1. Ідентифікація підозрілих компонентів – на основі результатів тестування, де виявлено некоректні вихідні сигнали, визначаються стовпці в таблиці несправностей, які відповідають потенційно несправним компонентам. Ці стовпці виділяються, оскільки вони пов'язані з вхідними тестовими наборами, що призвели до помилок.

2. Уточнення інформації про несправності – для виділених стовпців таблиці несправностей, значення, що відповідають вхідним сигналам тестових наборів, інвертуються. Це дозволяє більш точно визначити, які саме вхідні сигнали могли спричинити несправність.

3. Аналіз можливих комбінацій несправностей – використовуються логічні операції об'єднання (OR) та перетину (AND) над виділеними стовпцями таблиці несправностей. Об'єднання дозволяє виявити множинні несправності, а перетин – одиночні. Це дає можливість визначити як найгірший, так і найкращий сценарій щодо локалізації несправностей на вхідних лініях.

4. Локалізація несправностей на рівні лінії або елемента – за допомогою методу зворотного проходження по схемі, використовуючи дедуктивні матриці для кожного елемента, можна визначити місцезнаходження несправності з високою точністю, аж до окремої лінії або компонента.

На дедуктивній матриці можна вирішувати і обернене завдання, коли відомі дефекти і потрібно знайти тест, який їх перевіряє. Для цього потрібно від заданих дефектів таблиці істинності несправностей спуститися до одиниць матриці дедукції і потім кожної активної одиниці матриці знайти її проекцію на адресах таблиці істинності тестів.

3.3 Повна схема діагностування логічної функціональності

Схема діагностування вхідних несправностей за логічними векторами їх похідних. Синтез – це імплементація моделі у дозволену технологією систему логічних елементів для подальшого моделювання вхідних наборів, які не потребують алгоритму. Моделінг – це імплементація моделі в будь-яку пам'ять для подальшого моделювання вхідних наборів на основі транзакцій read-write без участі процесора. Враховуючи, що швидкодія пам'яті та ALU вже можна порівняти за швидкодією, синтез є не найкращим рішенням для побудови моделі. Тому далі розглядається моделювання схем, які використовують логічні вектори, розміщені в пам'яті, для діагностування будь-яких вхідних несправностей цифрової функціональності. Це краще, ніж виконувати синтез логічної схеми на основі нормальної диз'юнктивної форми, отриманої за одиничними значеннями логічного вектору. Слід зауважити, що за логічним вектором за три матричні операції (рис. 3.7) можна отримати картку тестування будь-якої функціональності.

Activity Matrix		Encoding matrices		Deductive Matrix								Testing Map																							
1	1	1	0	0	0	0	1									000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111				
1		1	1	1	1			0	1	2	3	4	5	6	7	000				1	1	1	1					000			.11	1..	1.1	11.	
1		1	1	1	1			1	0	3	2	5	4	7	6	001			1		1	1		1				001		.1		1..	1.0		110
1		1	1	1	1			2	3	0	1	6	7	4	5	010		1			1		1	1				010	.1			1..		10.	101
0	1	1	1			1		3	2	1	0	7	6	5	4	011		1	1	1	1							011	.0	.0	.00	1..			
0	1	1	1			1		4	5	6	7	0	1	2	3	100				1	1	1	1					100			.11	0..	0.1	01.	
0	1	1	1			1		5	4	7	6	1	0	3	2	101			1		1	1		1				101		.1		0..	0.0		010
0	1	1	1			1		6	7	4	5	2	3	0	1	110		1			1		1	1				110	.1			0..		00.	001
1		1	1	1	1			7	6	5	4	3	2	1	0	111		1	1	1	1							111	.0	.0	.00	0..			

Рисунок 3.7 – Дедуктивна матриця для логічного вектору 11100001

При цьому проміжним результатом є дедуктивна матриця, вектори якої використовуються для моделювання несправностей логічних елементів цифрової структури. Стівці дедуктивної матриці являють собою похідні, які також є логічними векторами і мають властивість пропускати всі сигнали, у тому числі, і несправності від зовнішнього входу до виходу елемента. Враховуючи, що дедуктивна матриця має 2^n стовпців (і рядків), які представляють всі можливі комбінації похідних щодо вхідних змінних. У даному випадку розглядаються лише n стовпців за кількістю змінних

функціональності (рис. 3.8). Розглядаються стовпці, які являють собою похідні за вхідними змінними

$$X'_1 = 11100001, X'_2 = 00110011, X'_3 = 11111111.$$

Ці стовпці формують три елементи в пам'яті, які відповідатимуть за виявлення та діагностування несправностей за відповідними входами.

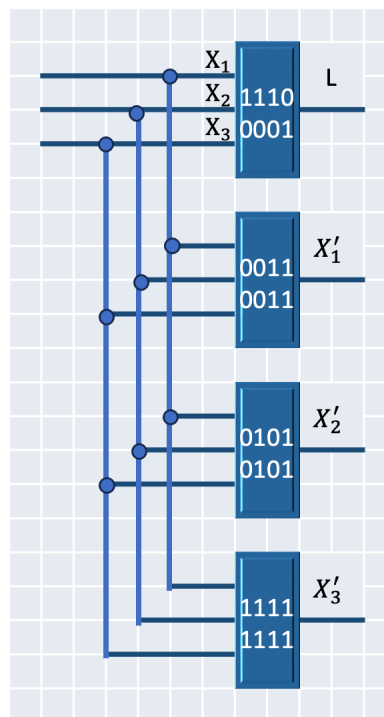


Рисунок 3.8 – Схема діагностування несправностей вхідних змінних

Якщо мати на увазі, що корисна функціональність містить логічний вектор такої ж довжини, виникає 300% надмірність схеми діагностування. Вона може бути виправдана лише можливістю діагностування та виправлення несправності там, де їхня поява дорого обходиться соціуму – у критичних системах, як транспорт, атомна енергетика, авіоніка. Слід зауважити, що булеві похідні виходять, як проміжний етап у процесі побудови карти тестування.

Основна ідея – побудова вичерпного тесту на логічному векторі функціональності за три матричні операції, а потім вилучення з цього тесту

будь-якої вхідної послідовності, яка в достатній мірі перевірятиме суттєві несправності.

Три елементи діагностування у схемі (див. рис. 3.8) можна об'єднати в один, який матиме три виходи за кількістю булевих похідних першого порядку. Якщо похідний стовпець дорівнює одиниці по всіх координатах – це означає, що несправність на вході (3) буде виявлена на всіх тестових наборах.

У більш загальному випадку схема діагностування будь-якої комбінації вхідних несправностей, яка побудована за дедуктивною матрицею функціональності, заданою логічним вектором 10000001, подана на рис. 3.9.

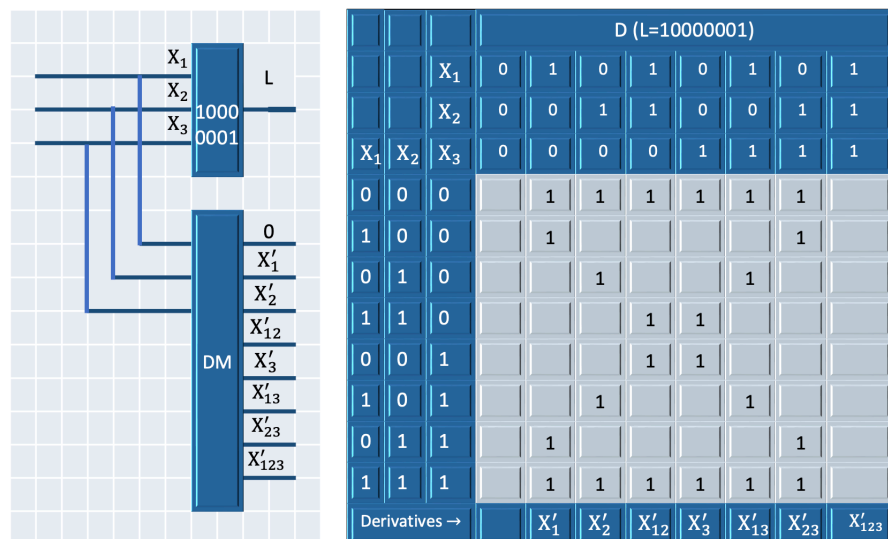


Рисунок 3.9 – Схема діагностування будь-якої комбінації несправностей

Природно, кількість виходів другого елемента DM визначається стовпцями дедуктивної матриці. Залежно від розв'язуваних задач можна вибирати від 1 до повної комбінації всіх стовпців дедуктивної матриці.

Для цієї функціональності суттєвих стовпців буде шість, що формує шість входів елемента DM. Перший та останній вихід не бере участі у діагностуванні комбінацій вхідних дефектів. Це впливає з дедуктивної матриці, де ці два стовпці є нульовими за всіма координатами.

3.4 Висновки до розділу 3

Слід зауважити, що схема діагностування є гнучким механізмом, де можна використовувати також один вихід, який поєднуватиме всі похідні або стовпці дедуктивної матриці. Тут все залежить від вимог вирішуваного завдання діагностування. Зазвичай обмежуються числом виходів у DM елементі, рівним числу входів корисної функціональності.

ВИСНОВКИ

1. Проаналізовано технологічні тенденції, що визначені компанією Gartner. Актуальними для комп'ютерної інженерії є розвиток енергоефективних обчислень (розробка та експлуатація комп'ютерів), гібридних обчислень як комбінування різних обчислювальних технологій – центральні процесори (CPU), графічні процесори (GPU), спеціалізовані мікросхеми (ASIC), квантові комп'ютери, а також просторові обчислення. У зв'язку з цим питання діагностики обчислювальних пристроїв є затребуваними.

2. Проаналізовано математичний апарат та технологічні рішення векторно-логічного комп'ютингу, значний внесок до розвитку теорії якого зробили вчені кафедри АПОТ, що відзначено у роботах [3-10].

3. Виконано синтез схем діагностики вхідної змінної на основі теорії векторно-логічного комп'ютингу: розглянуто синтез схем діагностування несправностей вхідних змінних на основі всіх булевих похідних логічної функції; наведено приклади синтезу схем діагностування одиночних несправностей вхідних сигналів для логічної функції з використанням дедуктивної матриці; на основі дедуктивної матриці синтезовано чотири схеми транспортування несправностей на вихід схеми, що дозволяють виявляти несправності під час роботи пристрою; узагальнено схему діагностування будь-якої комбінації вхідних несправностей, яка побудована за дедуктивною матрицею функціональності, заданою певним логічним вектором.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. 2025 Gartner Top 10 Strategic Technology Trends. Джин Альварес (Gene Alvarez), Gartner. October 10, 2024. [<https://www.gartner.com/en/articles/top-technology-trends-2025>]
2. Spotlight on 2024 Gartner Hype Cycle™ for Emerging Technologies Disruptive technologies hold great potential – for those businesses capable of overcoming the risks involved. Arun Chandrasekaran. October 10, 2024. [<https://www.gartner.com/en/articles/hype-cycle-for-emerging-technologies>]
3. Vector-deductive memory-based transaction for fault-as address simulation / [W. Gharibi, A. Hahanova, V. Hahanov, S. Chumachenko, E. Litvinova, I. Hahanov] // Electronic modeling. – 2023. – №45(1). – P. 3-26. <https://doi.org/10.15407/emodel.45.01.003>
4. Vector-Logic Synthesis of Deductive Matrices for Fault Simulation / [W. Gharibi, A. Hahanova, V. Hahanov, S. Chumachenko, E. Litvinova, I. Hahanov] // Electronic modeling. – 2023. – №45(2). – P. 16-33. <https://doi.org/10.15407/emodel.45.02.016>
5. Qubit Test Synthesis Processor for SoC Logic / [W. Gharibi, D. Devadze, V. Hahanov, E. Litvinova, I. Hahanov] // 2019 IEEE East-West Design & Test Symposium (EWDTS). – Batumi, Georgia. – 13-16 September 2019: proceedings. – IEEE, 2019. – P. 1-5. doi: 10.1109/EWDTS.2019.8884476.
6. Test Synthesis for Logical X-functions / [V. Hahanov, M. Liubarskyi, W. Gharibi, S. Chumachenko, E. Litvinova, I. Hahanov] // 2018 IEEE East-West Design & Test Symposium (EWDTS). – 14-17 September 2018: proceedings. – IEEE, 2018. – P. 1-9, doi: 10.1109/EWDTS.2018.8524863.
7. Vladimir Hahanov. Cyber-Physical Computing for IoT-driven Services. – New York: Springer, 2018. – 279 p. <https://doi.org/10.1007/978-3-319-54825-8>

8. Devadze D. Vector-Deductive Memory-Based Transactions for Fault-As-Address Simulation / D. Devadze, Z. Davitadze, A. Hahanova // 12th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 2022: proceedings. – IEEE, 2022. – P. 1–6. <https://doi.org/10.1109/dessert58054.2022.10018769>
9. Векторно-логічне моделювання несправностей / В.І. Хаханов, С.В. Чумаченко, Є.І. Литвинова та ін. // Радіоелектроніка, інформатика, управління. – Запоріжжя: ЗНТУ. – 2023. – №2 (65). – С.37–51. <https://doi.org/10.15588/1607-3274-2023-2-5>
10. Hahanova A. Vector-Deductive Faults-as-Address Simulation / A. Hahanova // International Journal of Computing. – 2023. – №22(3). – P. 328–334. <https://doi.org/10.47839/ijc.22.3.3227>
11. Skobtsov Y. A. Distributed Genetic Algorithm of Test Generation For Digital Circuits // Y. A. Skobtsov, A. El-Khatib, D. E. Ivanov // 2006 International Biennial Baltic Electronics Conference. – Tallinn, Estonia. – 02-04 October 2006: proceedings. – IEEE, 2006. – P. 1-4. doi: 10.1109/BEC.2006.311096.
12. Mrena M. Efficient Computation of Logic Derivatives Using Multi-valued Decision Diagrams / M. Mrena and M. Kvassay // 2024 IEEE 54th International Symposium on Multiple-Valued Logic (ISMVL). – Brno, Czech Republic. – 28-30 May 2024: proceedings. – IEEE, 2024. – P. 1-4. 143-148. doi: 10.1109/ISMVL60454.2024.00036.
13. Generic Constructions of (Boolean and Vectorial) Bent Functions and Their Consequences / [Y. Li, H. Kan, S. Mesnager, J. Peng, C. H. Tan, and L. Zheng] // IEEE Transactions on Information Theory. – 2022. – Vol. 68, no. 4. – P. 2735-2751. doi: 10.1109/TIT.2022.3140180.