

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)

Система діагностування несправностей за картою тестування  
(тема)

Виконав: здобувач IV курсу, групи КІУКІ-21-9  
Цівенко О.О.  
(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 Цівенко Олександр Олексійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Система діагностування несправностей за картою тестування  
затверджена наказом по університету від " 21 " 05 2025 р. № 403 Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 17.06.2025

3. Вихідні дані до роботи (проекту) \_\_\_\_\_

Моделі, метрики, технологічні рішення векторно-логічного комп'ютингу

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

Огляд технологій та публікацій

Аналіз моделей, методів, структур даних, технологій векторно-логічного комп'ютингу

Розробка системи діагностування несправностей за картою тестування.

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

17 слайдів

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 моделей, методів та алгоритмів	10.05.2025 - 17.05.2025	
4	Реалізація процедур для діагностування	18.05.2025 - 30.05.2025	
5	Синтез карт тестування діагностування несправностей за картою тестування	01.06.2025 - 05.06.2025	
6	Оформлення пояснювальної записки	06.06.2025 - 10.06.2025	
7	Перевірка виконаного проекту керівником,	11.06.2025 - 12.06.2025	
8	Захист проекту	19.06.2025	

Дата видачі завдання 06.05.2025

Здобувач \_\_\_\_\_



(підпис)

Цівенко О.О.

Керівник роботи \_\_\_\_\_



(підпис)

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

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 45 с., 12 рисунків, 1 дод., 14 джерел.

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

Тематика роботи стосується питань розробки системи діагностування несправностей за картою тестування на основі моделей, метрик та технологій векторно-логічного комп'ютингу.

В роботі розглянуто огляд стану технологій, аналіз моделей, методів, метрик, технологій векторно-логічного комп'ютингу, розробка системи діагностування несправностей за картою тестування.

## ABSTRACT

Bachelor's thesis contains 45 pages format A4, 12 figures, 1 application, 14 sources.

SYSTEM, VECTOR LOGIC COMPUTING, METRICS, TESTING MAP, DIAGNOSTICS, FAULT

The topic of the work concerns the development of a fault diagnosis system based on a test map based on models, metrics and technologies of vector logic computing.

The work reviews the state of the art, analyzes models, methods, metrics, technologies of vector logic computing, and develops a fault diagnosis system based on a test map.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 СТАН ТЕХНОЛОГІЙ	10
1.1 Топ тренди Gartner 2024	10
1.2 Метрика промислової революції 5.0	11
1.3 Аналіз літератури	12
1.4 Висновки та постановка завдання	23
2 МОДЕЛІ, МЕТОДИ, ТЕХНОЛОГІЧНІ РІШЕННЯ ВЕКТОРНО- ЛОГІЧНОГО КОМП'ЮТИНГУ	24
2.1 Векторно-логічний комп'ютинг	24
2.2 Моделі несправностей логічних схем	24
2.3 Структури даних	26
2.4 Векторно-матричний синтез карти тестування	29
2.5 Тестування помилок 1-го та 2-го роду	31
2.6 Висновки до розділу 2	33
3 СИСТЕМА ДІАГНОСТУВАННЯ НЕСПРАВНОСТЕЙ ЗА КАРТОЮ ТЕСТУВАННЯ	35
3.1 Діагностування дефектів за картою тестування	35
3.2 Процедура синтезу мінімального тесту за картою тестування	36
3.3 Процедура визначення відношення включення	36
3.4 Процедура мінімізації рядків карти	38
3.5 Процедура покриття несправностей тестовим набором	39
3.6 Алгоритм розв'язку задачі покриття	40
3.7 Синтез карти тестування від трьох змінних	41
3.8 Процедура моделювання несправностей за таблицею істинності	43
3.9 Висновки до розділу 3	44

	7
ВИСНОВКИ	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	46
ДОДАТОК А_Графічна частина	48

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence (штучний інтелект);

AI-TRiSM – AI Trust, Risk, and Security Management;

DToC – Digital twin of the customer (цифровий двійник клієнта);

GenAI – Generative AI (генеративний ШІ);

IoT – Internet of Things;

ШІ – штучний інтелект.

## ВСТУП

Тематика роботи стосується питань розробки системи діагностування несправностей за картою тестування на основі моделей, метрик та технологій векторно-логічного комп'ютингу.

Задачі: огляд стану технологій; аналіз сучасних публікацій; аналіз актуальних моделей, методів, технологій векторно-логічного комп'ютингу; розробки системи діагностування несправностей за картою тестування.

Мета бакалаврської роботи – зниження витрат при моделюванні несправностей логічних схем шляхом розробки системи діагностування несправностей на основі тестової картки з використанням векторно-логічних обчислювальних механізмів.

## 1 СТАН ТЕХНОЛОГІЙ

Аналізується стан сучасних технологій на основі прогнозів консалтингової компанії Gartner для топ-трендів Hype Emerging Technologies Cycle 2024 року. Наводиться огляд літератури.

### 1.1 Топ тренди Gartner 2024

Gartner Inc., яка прогнозує глобальну технологічну кібермоду, додала [1] 10 нових трендів до свого бренду Hype Emerging Technologies Cycle (рис. 1.1): суперкомп'ютери на основі штучного інтелекту, автономні агенти, людиноподібні робочі роботи, хмарні технології, програмна інженерія з доповненим штучним інтелектом, оперативна інженерія, 6G для надвисоких швидкостей, низької затримки, такі додатки, як імерсивна доповнена/віртуальна реальність та взаємодія штучного інтелекту в реальному часі, цифровий двійник клієнта (DToC), просторові обчислення інтегрують фізичне та цифрове середовища, дозволяючи користувачам взаємодіяти з цифровим контентом у 3D-просторі.

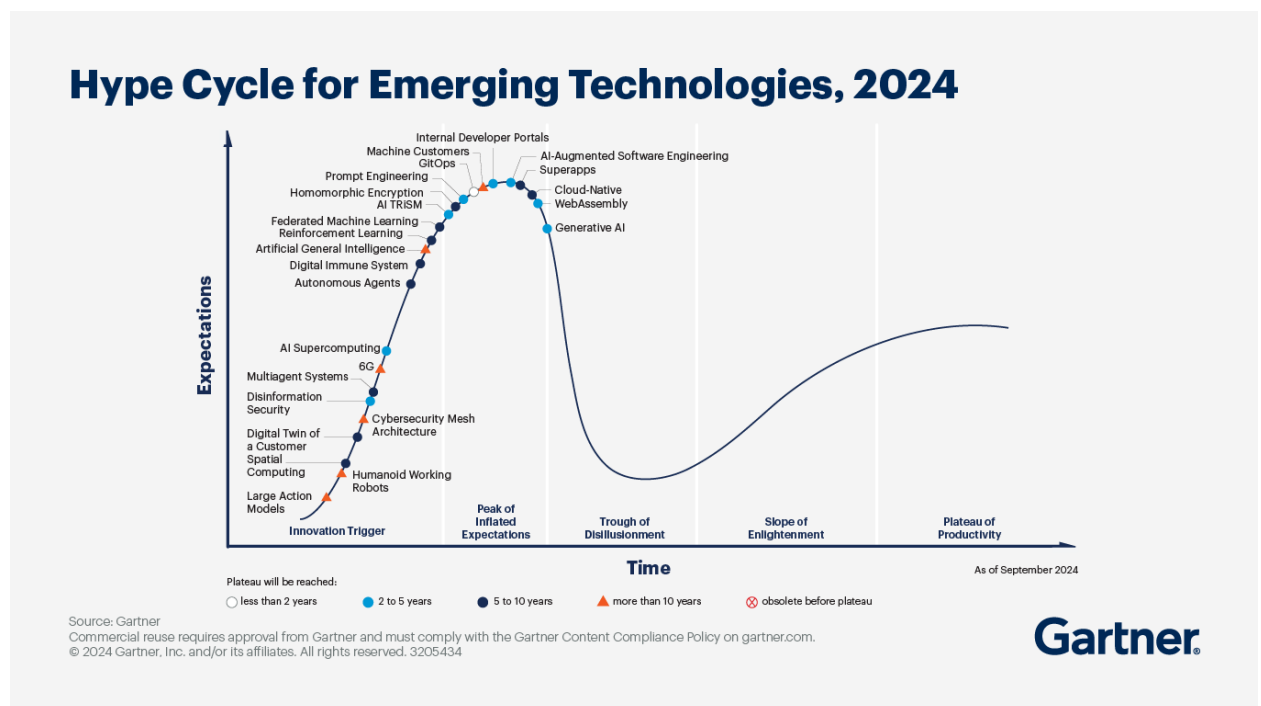


Рисунок 1.1 – 10 нових трендів Gartner Hype Emerging Technologies Cycle [1]

Автономні агенти – це обчислення, які досягають цілей без втручання людини. Це автомат, який використовує методи штучного інтелекту для визначення моделей навколишнього середовища, прийняття рішень, виконання дій та генерації результатів.

Gartner висуває найважливіший тренд в управлінні персоналом – залучення, утримання та розвиток талантів.

AI-Gartner пропонує нову парадигму штучного інтелекту для компаній та університетів: технологічний «сендвіч» зі штучним інтелектом, який зосереджений на зборі неструктурованих даних (80%) з усіх джерел.

Gartner пропонує масштабування механізмів штучного інтелекту: вбудований штучний інтелект (вбудований ШІ), вбудований ШІ та штучний інтелект «принеси свій власний», що працюють під захистом технологій AI-TRiSM [2].

## 1.2 Метрика промислової революції 5.0

Метрика промислової революції 5.0 поєднує сильні сторони креативних людей, їхню адаптивність, здатність вирішувати проблеми та емоційний інтелект з точністю та ефективністю комп'ютерних систем і робототехніки. Цей підхід визнає обмеження чистої автоматизації та штучного інтелекту, особливо у складних процесах прийняття рішень, інноваціях та контекстах, що вимагають емпатії та людського судження. Індустрія 5.0 сприяє інтеграції працівників у виробничі процеси разом із передовими технологіями штучного інтелекту для підвищення продуктивності, якості продукції та загальної конкурентоспроможності. Двигуни штучного інтелекту передбачають майбутнє, де люди та машини працюють разом у гармонії, кожен зі своїм унікальним внеском та сприяють більш гнучкому, сталому та адаптивному промислового та навчального ландшафту [3].

### 1.3 Аналіз літератури

Оглядова стаття [3] пропонує комплексний аналіз автоматизації розумної освіти в контексті Індустрії 5.0 на основі 78 статей, зосереджуючись на інтеграції передових технологій та розробці інноваційних, ефективних та етичних освітніх рішень для майбутніх фахівців. Оскільки світ переходить в епоху, що характеризується співпрацею людини та машини та технологіями, що швидко розвиваються, існує нагальна потреба визнати ключову роль розумної освіти в підготовці людей до можливостей та викликів, що виникають у новому промисловому ландшафті. У статті розглядаються ключові компоненти розумної освіти, включаючи інтелектуальні системи репетиторства, адаптивні навчальні середовища, аналітику навчання та застосування Інтернету речей (IoT) в освіті. Обговорюється роль передових технологій, таких як штучний інтелект (ШІ), машинне навчання (МН), робототехніка, а також доповнена та віртуальна реальність (AR/VR), у формуванні персоналізованого та захопливого навчального досвіду. В огляді підкреслюється важливість розумної освіти у задоволенні зростаючого попиту на підвищення кваліфікації та перекваліфікацію, сприянні культурі навчання протягом усього життя та сприянні адаптивності, стійкості та самовдосконаленню серед учнів. Крім того, у статті заглиблюються у проблеми та етичні міркування, пов'язані з впровадженням розумної освіти, розглядаючи такі питання, як конфіденційність даних, цифровий розрив, готовність вчителів та учнів, а також потенційні упередження в системах на базі штучного інтелекту. За допомогою презентації тематичних досліджень та прикладів успішних ініціатив у сфері розумної освіти, огляд має на меті надихнути освітян, політиків та зацікавлені сторони галузі на співпрацю та впровадження інновацій у розробці та впровадженні ефективних рішень для розумної освіти. Окреслюються нові тенденції, майбутні напрямки та потенційні дослідницькі можливості в галузі розумної освіти, підкреслюючи

важливість постійного вдосконалення та інтеграції нових технологій для забезпечення актуальності та ефективності освіти в контексті Індустрії 5.0. Надаючи цілісне розуміння ключових компонентів, проблем та потенційних рішень, пов'язаних з розумною освітою, ця оглядова стаття прагне зробити внесок у поточну дискусію щодо автоматизації розумної освіти та її ролі в підготовці здобувачів до майбутнього праці.

В статті [4] основною ідеєю є розробка підходу до векторного комп'ютингу, який усуває залежність від традиційної логіки, застосовуючи лише операції читання-запису в адресній пам'яті. Головна стратегічна мета полягає у створенні детермінованих векторно-квантових обчислень із залученням фотонів для виконання транзакцій читання-запису на стабільних елементах субатомної пам'яті. Основним завданням виступає впровадження нових моделей і методів векторних обчислень, базованих на примітивних транзакціях читання-запису в технології гнучкого інтерпретативного моделювання та симуляції помилок, де дані одночасно виконують роль адрес для обробки самих себе. Сутність такого підходу обчислень полягає у використанні векторних структур даних в операціях читання-запису із пам'яті. Векторні обчислення являють собою процес, що базується на простих транзакціях читання-запису над двійковими векторами, які зберігаються в адресній пам'яті та забезпечують функціональність, де самі адреси комірок є вхідними даними для подальшої обробки. Виявлені переваги універсальної векторної моделі, що забезпечує компактний опис упорядкованих процесів, явищ, функцій та структур з можливістю їх паралельного аналізу. Аналітична логіка, яка раніше потребувала складних алгоритмічних обчислень, замінюється вихідними векторами станів елементів і цифровими схемами, орієнтованими на паралелізм операцій у регістровій логіці на регулярних структурах даних. Запропоновано додатково векторно-дедуктивний метод синтезу формул для поширення вихідних списків несправностей, який характеризується квадратичною складністю виконання регістрових операцій. Розроблено нову матрицю дедуктивних

векторів із низкою важливих властивостей: компактністю, можливістю паралельної обробки даних через одну транзакцію читання-запису, виключенням традиційної логіки з процесів симуляції несправностей, повною автоматизацією синтезу та спрямуванням на технологічне вирішення завдань технічної діагностики. Робота також передбачає перехід до використання векторної логіки для організації обчислень із повним виключенням старої моделі логіки у вигляді таблиць чи аналітичних виразів. Застосування виключно транзакцій читання-запису у пам'яті без необхідності використання традиційних команд відкриває можливості для розвитку детермінованих квантових архітектур, побудованих на стабільних субатомних частинках пам'яті.

Основна ідея статті [5] полягає у розробці векторно-логічних обчислень, які базуються виключно на транзакціях читання-запису адресної пам'яті для обробки великих масивів даних. Головною метою є створення нових простих і надійних моделей та методів виконання векторних обчислень, що опираються на примітивні транзакції читання-запису в межах технології векторного гнучкого інтерпретативного моделювання несправностей цифрових систем. Векторно-логічні обчислення представляють собою процес обробки, заснований на маніпуляції бітами двійкового вектора функціональності через операції читання-запису, в яких адреси бітів виступають вхідними даними. У статті запропоновано векторний метод синтезу дедуктивних матриць для передачі вхідних списків несправностей, що має квадратичну обчислювальну складність. Цей метод розвиває підхід алгоритмічного дедуктивного синтезу векторів, побудованого на таблицях істинності. Дедуктивна матриця використовується для синтезу та перевірки тестів із застосуванням технології паралельного моделювання несправностей. Цей процес базується на виконанні транзакцій читання-запису дедуктивних комірок пам'яті, що утворюють вектори тестових даних.

В роботі [6] розроблено кубітний метод синтезу тестів дискретних функцій компонентів SoC, який базується на використанні булевих похідних у контексті векторного опису роботи логічного елемента у вигляді Q-покриття. Визначено пріоритетність метрик математичних і технологічних співвідношень у структурах даних, які слугують основою для побудови ефективних алгоритмів управління або обробки даних, спрямованих на підвищення продуктивності процесів тестування. Запропоновано векторну модель булевих похідних, що використовується як для синтезу дедуктивних матриць у моделюванні відмов кубітів, так і для оцінки якості тестових послідовностей. Також розроблено деревоподібний ATPG-процесор, виконаний у формі бінарного графа на основі xor-елементів, який забезпечує паралельну обробку частин покриття кубітів. Крім того, запропоновано структури даних для логіки SoC, що використовуються для розрахунку булевих похідних кубітів. Усі описані структури та методи реалізовані у вигляді програмного застосунку, орієнтованого на паралельне тестування логічних функцій цифрових систем на кристалах із використанням технології покриття кубітів.

У [7] представлено клас логічних функцій, зокрема X-функцій (xor, not-xor), разом із кубітними моделями, які є практичним інструментом для тестування, діагностики та моделювання несправностей компонентів системи на кристалі (SoC). Запропоновано кубітні моделі, а також методи для моделювання та аналізу цифрових пристроїв і їхніх компонентів. Розглянуто паралельні підходи до мінімізації логічних функцій, виявлення несправностей у SoC та вирішення завдань покриття через унітарне кодування кубітних структур даних. Описується архітектура сервісів, призначених для проектування, тестування та верифікації цифрових пристроїв на основі кубітних моделей логічних примітивів. Також розглянуто сервіс для моделювання відмовостійких схем із використанням кубітного покриття функціональних примітивів. Усі моделі, кубітні структури даних і методи орієнтовані на реалізацію і дослідження на

класичних комп'ютерах із застосуванням унітарного кодування двійкових станів.

Наукова монографія [8] описує автоматизовану обчислювальну модель, яка розробляє і пояснює технологічний моніторинг та управління різноманітними процесами й явищами у фізичному, віртуальному та космологічному просторах. Вона також зосереджена на формулюванні теорій та технологій, що базуються на використанні кубітів для оптимізації проєктування, тестування, верифікації та діагностики систем із пам'яттю. Розглядається кіберкультура як сукупність соціальних і технологічних взаємозв'язків між суспільством, фізичним середовищем і кіберпростором, що спрямовані на цифровий моніторинг і управління ресурсами хмарних обчислень, зокрема у сферах освіти і транспорту. Пропонується новий метод із використанням булевої метричної логіки, створений для асоціативної обробки даних без застосування арифметичних операцій. Крім того, описується автоматизована космологічна модель, яка детально пояснює циклічну та гармонійну еволюцію матерії, енергії, простору й часу в контексті розвитку Всесвіту.

Ідея роботи [9] полягає у створенні вільних від логіки векторних обчислень, використовуючи лише транзакції читання-запису в адресній 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. Введено метричну аксіому згорткового замикання циклічних відстаней між  $n$  об'єктами в  $0$ -простір. Запропоновано модель хог-відношень

між чудовими логічними функціями ( $i \oplus$  або  $\oplus \text{ xor} = 0$ ) цифрових схем; вона згорнута в нульовий простір, що дозволяє вирішувати задачі технічної діагностики, генеративного машинного навчання, пошуку подібності-різниці між процесами та явищами. Введено метрику управління, керовану відмовами,  $T \oplus F \oplus L = 0$ , яка формалізує всі відомі процеси створення обчислень, включаючи проектування та тестування, кіберфізичні та кіберсоціальні обчислення, федеративні та генеративні ML-обчислення. Введено рівняння логічного аналізу  $S \oplus D = aUb = U$ , яке дозволяє обчислювати подібність-різницю між об'єктами за допомогою паралельних логічних процедур на бінарних векторах, що утворюють U-всесвіт. 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, замінено 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. Запропоновано нову структуру секвенсора векторного дедуктивного моделювання несправностей, яка характеризується простотою реалізації на одному блоці пам'яті, виключенням будь-якої традиційної логіки, використанням транзакцій читання-запису даних у пам'яті для генерації вихідного вектора несправностей, використанням даних як адрес для самої обробки даних.

Основна ідея статті [10] зосереджена на створенні векторно-логічних обчислень у оперативній пам'яті (VLC), що використовують виключно операції читання-запису в адресній пам'яті для моделювання несправностей як адрес. Традиційна логіка при цьому не застосовується. VLC не потребує залучення команд процесора чи АЛП, що робить його ідеальним для реалізації в SoC та FPGA. Запропоновано векторно-логічний метод дедуктивного матричного синтезу для перенесення вхідних несправностей, який демонструє квадратичну обчислювальну складність. Також розроблено симулятор-автомат для оперативної пам'яті, що дозволяє виконувати векторно-дедуктивне моделювання несправностей на основі транзакцій читання-запису з перспективою реалізації в SoC. Мета. Розробити векторно-дедуктивний метод моделювання несправностей із застосуванням примітивних транзакцій читання-запису для аналізу логічних схем. Метод. Використано вхідний тестовий набір та вектор функціональної логіки. Метод базується на алгоритмі синтезу дедуктивних векторів за допомогою таблиць істинності. Дедуктивна матриця застосовується для синтезу та верифікації тестових сигналів через паралельне моделювання комбінацій несправностей, представлених як адреси. Операції читання-запису виконуються над бітами дедуктивних векторів у пам'яті. Результати. Розроблено методику синтезу дедуктивних матриць для транспортування вхідних несправностей до вихідних сигналів елемента. Визначені структури даних, що підтримують паралельне моделювання несправностей цифрових схем на основі транзакцій читання-запису в матричній пам'яті, де стовпці адрес представляють комбінації несправностей. Створено секвенсор із п'яти блоків, що забезпечує реалізацію векторно-логічних обчислень, пов'язаних із дедуктивним моделюванням несправностей на базі транзакцій читання-запису. Проведено верифікацію запропонованих моделей та методів на тестових прикладах. Робота містить низку наукових інновацій, зокрема: 1) уперше запропоновано векторно-логічний метод синтезу матриць дедуктивних векторів для паралельного моделювання комбінацій вхідних несправностей як адрес; 2)

уперше розроблено автомат для векторно-дедуктивного моделювання несправностей як адрес із використанням транзакцій читання-запису, оптимізований для реалізації на базі LUT FPGA та інтегрованого онлайн-симулятора SoC для рівня RTL; 3) продемонстровано переваги технології векторно-логічного синтезу дедуктивних матриць на прикладах традиційної та RTL-логіки, що підкреслює ефективність цього підходу порівняно з аналітичними формулами під час створення симуляторів; 4) використання матриці дедуктивних векторів у вигляді набору вектор-стовпців булевих похідних дозволяє створювати мінімальні тести для логічних елементів; 5) застосування рекурсивної формули для синтезу матриці координатних перестановок у векторах активності спрощує отримання дедуктивної матриці для моделювання несправностей як адрес. Практична цінність полягає у забезпеченні високої швидкості моделювання несправностей цифрових блоків у SoC — до сотень наносекунд — при використанні симулятора в оперативній пам'яті. Надано оцінки складності відповідних алгоритмів.

Основна ідея роботи [11] полягає у створенні векторної моделі без використання традиційної логіки, заснованої виключно на транзакціях читання-запису в адресній пам'яті. Метод векторного моделювання несправностей застосовується як інструмент для оцінки якості тестів складних IP-ядер, реалізованих у технологіях FPGA та ASIC. Головне завдання – розробити прості та надійні моделі і методи векторних обчислень на основі базових транзакцій читання-запису, що підтримують технологію гнучкого інтерпретативного моделювання несправностей. У векторних обчисленнях проводяться операції над бітами двійкових векторів, де адреси бітів виступають вхідними даними. Запропоновано векторно-дедуктивний метод синтезу тестових векторів для аналізу списків несправностей, що відзначається квадратичним рівнем обчислювальної складності. Логічні вирази, які зазвичай потребують складної алгоритмічної обробки, замінюються вихідними станами елементів цифрових схем, представленими у вигляді векторів. Сформовано нову матрицю дедуктивних векторів, яка має

такі характеристики: компактність, можливість паралельної обробки даних на основі централізованої транзакції читання-запису, заміна звичних логічних процедур у процесах моделювання несправностей, повна автоматизація синтезу та здатність вирішувати широкий спектр діагностичних задач. Запропонована нова структура секвенсора для векторного дедуктивного моделювання несправностей. Вона відзначається простою архітектурною реалізацією на єдиному блоці пам'яті, усуває потребу у традиційній логіці, виконує формування вихідного вектора несправностей на основі транзакцій читання-запису та використовує дані як адреси для обробки цих же даних.

В статті [12] відзначається, що структури даних відіграють ключову роль у створенні ефективних паралельних алгоритмів і високопродуктивних обчислювальних пристроїв. У процесі розробки 80% часу присвячується створенню математично досконалих та технологічно простих структур даних, тоді як лише 20% витрачається на алгоритми та їх апаратно-програмне впровадження. Це стимулює пошук примітивів структур даних, які можуть суттєво спростити паралельні високопродуктивні алгоритми. Запропоновано моделі та методи тестування та моделювання цифрових систем з певними перевагами квантових обчислень щодо реалізації векторних кубітних структур даних у класичних обчислювальних процесах. Метою роботи є розробка технології кубіт-векторного синтезу та дедуктивного аналізу тестів для їх верифікації на основі векторних структур даних, що значно спрощують алгоритми і можуть бути вбудовані як компоненти BIST у мікросхемах цифрових систем. Дедуктивне моделювання несправностей застосовується для отримання аналітичних виразів, спрямованих на транспортування списків несправностей через функціональний або логічний елемент на основі операції хог, яка слугує мірою схожості або відмінності між тестом, функцією та несправностями, представленими у форматах таблиці, графіка чи рівняння. Бінарний вектор розглядається як найефективніший примітив структур даних для завдання логічної

функціональності з метою паралельного синтезу та аналізу цифрових систем. Паралелізм в розв'язанні комбінаторних задач є фізичною властивістю квантових обчислень, яка в класичних системах досягається за допомогою унітарно-кодованих структур даних завдяки додатковій пам'яті. Результати: 1) розроблено методологію аналітичного синтезу дедуктивної логіки для функціональних елементів на рівні вентилів і передачі регістрів; 2) запропоновано дедуктивний процесор для моделювання несправностей шляхом перенесення вхідних списків або векторів несправностей на зовнішні виходи цифрових схем; 3) описано кубіт-векторну форму логічного завдання та методи кубітного синтезу дедуктивних рівнянь для моделювання несправностей; 4) розроблено кубіт-векторний метод синтезу тестів, що використовує похідні, отримані за векторним покриттям логіки; 5) верифікацію моделей та методів проведено на тестових прикладах програмної реалізації структур та алгоритмів. Наукова новизна полягає у новій парадигмі технології синтезу дедуктивної RTL-логіки, заснованої на використанні метричних тестових рівнянь, формуючи векторну форму для опису структур. Це дозволяє застосовувати існуючі технології синтезу та аналізу тестів логічних схем для ефективного вирішення задач тестування графових структур і моделей кінцевих автоматів цифрових пристроїв. Практичну важливість підтверджено прикладами аналітичного синтезу дедуктивної логіки для елементів на рівні вентилів і передачі регістрів. Запропоновано дедуктивний процесор для моделювання несправностей як інструмент BIST, застосовуваний в онлайн-тестуванні й діагностиці несправностей цифрових систем на кристалах. Запропоновано кубіт-векторну форму опису цифрових систем, яка перевершує існуючі методи за показниками технологічності, компактності, швидкості та якості. Розроблено програмне забезпечення для реалізації основних сервісів тестування, моделювання та діагностики, яке використовується в навчальному процесі.

В [13] розроблено модель XOR-взаємозв'язків між ключовими логічними функціями цифрових схем, яка згортається у нульовий простір. Це

дозволяє вирішувати завдання технічної діагностики, машинного навчання, а також пошуку схожостей і відмінностей у процесах та явищах. Виявлено переваги використання векторної моделі для компактного й ефективного опису процесів, явищ, функцій та їхніх структур. Замість аналітичних логічних виразів, що потребують застосування ресурсоемних калькуляторів-аналітиків, запропоновано використовувати векторні структури даних для представлення функціональної логіки. Розроблено також векторно-дедуктивний метод синтезу формул, оптимізований для транспортування вхідних списків несправностей, який забезпечує квадратичну обчислювальну складність реєстрових операцій. Запропоновано координатно-векторну модель представлення несправностей, що не залежить від вхідних змінних і може ефективно застосовуватись для аналізу складних логічних схем під час оцінювання якості створених тестів.

В статті [14] розроблено векторний підхід для створення карти тестування та моделювання несправностей у логічних схемах. Цей метод дозволяє безпосередньо визначати всі несправності, які виявляються за допомогою тестових наборів, а також формувати тестові набори для виявлення конкретних несправностей. Основу методу складає суперпозиція інтелектуальних структур даних, включаючи дедуктивну матрицю  $D$  як похідну логічного вектора  $L$ , таблицю істинності тестів  $T$  і таблицю істинності несправностей  $F$ . Дедуктивна матриця виступає геном функціональності та центральним елементом механізму моделювання несправностей, забезпечуючи вирішення завдань тестування та верифікації. Метод базується на ключовій аксіомі: всі задіяні таблиці мають ідентичну форму та взаємодіють між собою через операцію згортки  $T \oplus L \oplus F = 0$ . Представлено універсальний дедуктивний реверсивний перетворювач для переходу між доменами «тест-несправності» та «несправності-тест», що охоплює логічні функції будь-якої розмірності. Основні функції цього перетворювача включають моделювання несправностей на основі тестових наборів ( $T \rightarrow F$ ) та синтез тестових наборів для виявлення конкретних

несправностей ( $F \rightarrow T$ ). Такий підхід може слугувати основою для створення сервісів генерації тестів і моделювання несправностей у системах на кристалі (SoC) з використанням IP-ядра відповідно до стандарту IEEE 1500 SECT. На базі дедуктивної матриці створюється карта тестування несправностей, яка пов'язує кожен тестовий набір із несправностями логічних вхідних ліній, що він здатен виявити. Така карта забезпечує уніфікований і ефективний підхід до тестування та аналізу несправностей у цифровій електроніці.

#### 1.4 Висновки та постановка завдання

Наведено тенденції розвитку сучасних технологій, що визначено консалтинговою компанією Gartner. Серед них напрямки, що залишають актуальними розвиток підходів до тестування, діагностування моделювання цифрових систем на кристалах.

Мета бакалаврської роботи – зниження витрат при моделюванні несправностей логічних схем шляхом розробки системи діагностування несправностей на основі тестової карти з використанням векторно-логічних обчислювальних механізмів.

Задачі: огляд стану технологій; аналіз сучасних публікацій; аналіз актуальних моделей, методів, технологій векторно-логічного комп'ютингу; розробка системи діагностування несправностей за картою тестування.

## 2 МОДЕЛІ, МЕТОДИ, ТЕХНОЛОГІЧНІ РІШЕННЯ ВЕКТОРНО-ЛОГІЧНОГО КОМП'ЮТИНГУ

Розглядаються актуальні моделі, методи, технології векторно-логічного комп'ютингу, що застосовується у подальшому для розробки системи діагностування несправностей за картою тестування [3-14].

### 2.1 Векторно-логічний комп'ютинг

Векторно-логічні обчислення як економічно ефективний механізм інтелектуальних обчислень в пам'яті використовують транзакції читання-запису для вирішення практичних задач аналізу та управління фізичними, соціальними та бізнес-процесами на основі моніторингу. Механізм розглядається як гармонійний зв'язок між моделлю та алгоритмом для вирішення задач пасивних та активних (керуючих) обчислень. Для будь-якої ситуації можна побудувати таку надлишкову модель, яка обнуляє алгоритм її аналізу для отримання рішення. Метою є значне скорочення витрат часу та енергії на управління процесами та явищами у фізичному, соціальному та цифровому світі. Засобом досягнення цієї мети є використання векторно-логічних механізмів обчислень в пам'яті, які значно спрощують алгоритми завдяки експоненціальній надлишковості структур даних. Розглянуто механізми обнулення обчислювальної складності алгоритму завдяки експоненціальній складності структур даних на основі векторної логіки.

### 2.2 Моделі несправностей логічних схем

Моделі несправностей логічних елементів формуються координатами одиниць адрес адрес бітів логічного вектора або таблиці істини [4-6], які проводяться в двійковому тесті, встановленому у зворотних значеннях вхідних змінних, які перекладають функціональність в альтернативний стан.

Альтернативні стани логічного елемента представлені наборами 1- та 0-координат у векторі функціональності.

Таблиця істинності – логічний вектор  $y \in$  найбільш універсальним механізмом вирішення всіх завдань тестувальних функцій, структур, шини даних. На будь-якому комп'ютері це повний набір компонентів, які потребують тестування та перевірки. На основі логічного вектора будується дедуктивна матриця та карта тестування будь-якої функціональності чи структури (рис. 2.1). Крім того, логічний вектор – це універсальна комп'ютеризована модель для аналізу будь-яких процесів чи явищ з метою управління, цікавими для ІТ-ринку.

x1	x2	x3	x4	y	D	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	F	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
					0000	1	1		1					1									0000	...1	..1	..	..	..	..	..	..	..	..	..	..	..	..	..
			1	1	0001	1	1		1		1	1	1		1	1	1	1	1	1	1	0001	...0	..1	..	..	..	..11	..10	1...	...	1.1	1.10	11..	11.0	111	1110	
		1	1	1	0010	1	1		1	1		1	1	1		1	1	1	1	1	1	0010	...1	..0	..	..	..1	..11	..101	1...	1..1	1.01	11..	11.1	110	1101		
		1			0011	1	1				1					1						0011	...0	..0	..	..	..	..	..100	...	...	1.00	...	...	...	...		
		1		1	0100	1	1	1	1			1	1	1	1	1			1	1	1	0100	...1	..1	..11	..0	..	..011	1...	1.1	1.1	1.11	1.11	10.1	101	1011		
		1		1	0101	1				1								1				0101	...0	..	..	..0	..	..010	...	...	...	10.0	...	...	...	...		
		1	1		0110		1				1									1		0110	...0	..	..	..0	..	..001	...	...	...	100	...	...	...	...		
		1	1	1	0111			1	1	1	1									1	1	0111	...00	..	..	..0.0	..00	...	...	...	0.11	01.1	011	0111	...	...		
1				1	1000	1	1	1	1	1	1	1	1			1		1	1	1	1	1000	...1	..1	..11	..1	..1	..11	..111	0...	...	0.10	01.0	010	...	...		
1			1		1001	1								1				1				1001	...0	..	..	..	..	...	...	...	0.01	01.0	...	...	...	...		
1		1			1010		1							1		1						1010	...0	..	..	..	..	...	...	...	0.01	01.0	...	...	...	...		
1		1	1		1011			1						1	1						1	1011	...00	..	..	..	..	...	...	0.0	0.0	...	...	0100	...	...		
1	1				1100				1									1	1			1100	...0	..	..	..	..	...	...	0.0	...	00.1	001	...	...	...	...	
1	1		1		1101				1								1				1	1101	...0.0	..	..	..	..	...	...	0.0	...	00..	...	0010	...	...		
1	1	1			1110				1								1				1	1110	...00	..	..	..	..	...	...	0.0	...	00..	...	0001	...	...		
1	1	1	1		1111				1							1	1	1	1	1	1	1111	...000	..	..	..	..	...	...	0.00	...	00.0	000	...	...	...	...	

Рисунок 2.1 – Модель несправностей логічних схем

Моделі несправностей логічних схем утворюються за допомогою одиничних координат основної діагоналі матриці моделювання, які сприймаються у бінарному тесті на зворотні значення вхідних, внутрішніх та вихідних ліній, що перетворюють виходи елементів у альтернативні умови.

Моделі тестів, логіки та несправності створюють згорткове співвідношення

$$T \oplus L \oplus F = 0 \quad (2.1)$$

у просторі логічного вектора або адреси таблиць істинності. Будь-який компонент тріади – це N-умова між двома іншими компонентами. Комбінації

1 біт у адресах таблиці істинності – це система шаблонів  $L$ , яка перетворюється на логічні несправності  $F$  під дією тестових бінарних наборів  $T$ :  $F=T\oplus L$ .

Кожна така несправність, що утворюється за допомогою ненульової адреси таблиці істинності, порушує закон функціонування логічного елемента або структури, якщо він перекладає стан логічного елемента в альтернативу. Дедуктивна матриця є ефективним механізмом синтезу всіх критичних несправностей, визначених адресами таблиці істинності, що впливає на функціонування логічного елемента або структури на вхідних наборах тестів.

Карта тестування – це механізм миттєвого моделювання всіх несправностей на вичерпному тесті в результаті створення розумних структур даних. Адреса – це утворюючий фактор in-методу комп'ютерингу, 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.

### 2.3 Структури даних

Логічне тестування – це механізм, який містить розумні структури даних і алгоритм хог для їх аналізу з метою автоматичного синтезу тесту і моделювання несправностей.

Розумні структури даних [6-8] – 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  $2^n$ .

Рівняння хог-згортки (конволюції) відстаней між трьома об'єктами (процесами) в бінарному векторному просторі  $F \oplus L \oplus T = 0$  є слідством Н-метрики

$$\bigoplus_{i=1,n} \mathbf{d}_i = \mathbf{0}. \quad (2.2)$$

Відстань між будь-якими двома компонентами тріади F, L, T дорівнює третій компоненті.

Логічна матриця L – це впорядкована множина логічних векторів або їх інверсій:

$$\mathbf{L}_i = \mathbf{L}_j \oplus \mathbf{L}. \quad (2.3)$$

Логічна матриця L є декартовим хог-квадратом логічного вектора:  $\mathbf{L} = \mathbf{L} \oplus \mathbf{L}$ .

Матриця несправностей  $F = T \oplus L$  – це повний набір десяткових кодів (адрес-відстаней) або комбінації всіх логічних несправностей, які можна перевірити на вичерпному наборі тестів (запропоновано три способи побудови).

Матриця тестів T – це повний набір тестів, які вказують на несправності, які потрібно перевірити на кожному двійковому наборі. Вона визначається рівнянням  $T = L \oplus F$ . За допомогою десяткових кодів F-матриці можна спростити процедуру визначення несправностей, що перевіряються на тесті:  $T = \mathbf{L}_F$ .

Відомо, що повний тест логічної функціональності перевіряє наявність всіх несправностей, але не відомо, які саме несправності перевіряє кожен набір тестів. Це завдання можна вирішити за допомогою синтезу тестової карти на розумних структурах даних.

Матриця тестів на несправність  $F$   $T$  може бути перетворена на чітку карту випробувань несправностей  $T$ - $F$  за допомогою двох проекцій на вісь  $t$  і  $f$  кожної координати одиничної матриці  $T$ :

$$F_{ijt} = \bar{t}_{it} \leftarrow T_{ij} = f_{jt} = 1, \{I, j\} = 1, 2^n; t = 1, n; \quad (2.4)$$

$f_{ij}$  – розряди адрес таблиці несправностей;  $t_{ij}$  – розряди адрес таблиці тестів;  $T_{ij}$  – координати  $T$ -матриці,  $F_{ijt}$  – координати векторів  $F$ -матриці.

За дві матричні операції вирішуються задачі синтезу тестів (рис. 2.2), моделювання несправностей, тестування і перевірки будь-яких функціональних можливостей, заданих логічним вектором.

	L-matrix	F-matrix	F-matrix (D)	F-matrix (test-map)
L	0 0 0 1 1 0 1 1		000 001 010 011 100 101 110 111	000 001 010 011 100 101 110 111
0	1 1 1 1	0 1 2 3 4 5 6 7	000	000 .11 1.. 11. 111
0	1 1 1 1	1 0 3 2 5 4 7 6	001	001 .1. 1.0 11. 110
0	1 1 1 1	2 3 0 1 6 7 4 5	010	010 ..1 1.. 1.1 10.
1 1 1 1	1	3 2 1 0 7 6 5 4	011	011 ..0 .0. .00 10.
1 1 1 1	1	4 5 6 7 0 1 2 3	100	100 ..1 0.. 0.1 01.
0	1 1 1 1	5 4 7 6 1 0 3 2	101	101 ..0 .1. .10 01.
1 1 1 1	1	6 7 4 5 2 3 0 1	110	110 .01 0.. 00. 001
1 1 1 1	1	7 6 5 4 3 2 1 0	111	111 .0. 0.0 00. 000

Рисунок 2.2 – Синтез карти тестування

Матриця несправностей є константою, інші матриці – змінними. Константа  $F$  виходить для будь-яких значень тестів  $T$  і логічного  $L$ :

$$F = T \oplus L. \quad (2.5)$$

Формула автоматичного синтезу тесту та моделювання несправностей:

$$T = L \oplus F. \quad (2.6)$$

Формула синтезу логічного вектору з будь-якими заданими властивостями

$$L=T\oplus F. \quad (2.7)$$

Рівняння хог-згортки (конволюції) відстаней у двійковому векторному просторі (2.1) є наслідком H-метрики (2.2) для розв'язування задачі моделювання несправностей логіки. Відстань між будь-якими двома компонентами тріади (FLT) дорівнює третій компоненті. Це дає можливість вирішити три завдання на одній моделі design and test. Logic Testing – це хог-відношення трьох матриць: логіки L, несправностей F, тестів T, розмірністю  $2^n \times 2^n$ , n – кількість змінних. Всі матриці виходять з логічного вектору, у тому числі й F-матриця, яка є картою тестування логіки.

#### 2.4 Векторно-матричний синтез карти тестування

Процес синтезу базується на чотирьох матричних операціях з логічним вектором і його розширеннями (таблицями істинності), що дозволяє сформувати матрицю D-(F) (рис. 2.3).

L-matrix				H-matrix				D-matrix				X-matrix				F-matrix (test-map)													
0 0 0 1 1 0 1 1								000 001 010 011 100 101 110 111				TT 000 001 010 011 100 101 110 111				000 001 010 011 100 101 110 111													
0		1 1	1 1	0	1	2	3	4	5	6	7	000		1	1	1	1	000		.11	1..	11. 111	000		.11	1..	11. 111		
0		1 1	1 1	1	0	3	2	5	4	7	6	001		1		1	1	001		.1.		1.1 11. 111	001		.1.		1.0 11. 110		
0		1 1	1 1	2	3	0	1	6	7	4	5	010		1		1	1	010		..1		1.. 1.1 11.	010		..1		1.. 1.1 10.		
1	1	1	1	1		3	2	1	0	7	6	011		1	1	1	1	011		..1	.1.	..11	11.	011		..0	..0	..00	10.
1	1	1	1	1		4	5	6	7	0	1	100		1		1	1	100		..1		1.. 1.1 11.	100		..1		0.. 0.1 01.		
0		1 1	1 1	5	4	7	6	1	0	3	2	101		1	1	1	1	101		..1	.1.	..11	11.	101		..0	..1	..10	01.
1	1	1	1	1		6	7	4	5	2	3	110		1	1	1	1	110		..11	1..	11. 111	110		..01	0..	00. 001		
1	1	1	1	1		7	6	5	4	3	2	111		1		1	1	111		.1.		1.1 11. 111	111		..0.		0.0 00. 000		

Рисунок 2.3 – Схеми синтезу карти тестування логічної функціональності 00011011

На основі цієї матриці здійснюється моделювання (діагностування) логічних дефектів вхідних параметрів у паралельному режимі обробки TF (FT) векторів. Обчислювальна вартість покоординатного формування

матриці активності становить  $5 \times 2^n \times 2^n$ . Загалом, обсяг необхідної пам'яті –  $5 \times 2^n \times 2^n$ . Подібних рішень, що вирізняються простотою та прогресивністю, у світі немає. Основна складність полягає у представленні результатів моделювання (діагностування) логічних функцій великої розмірності. Витрати пам'яті та обчислювальна інтенсивність синтезу є прогнозованими.

Розглядаються наступні поняття.

1. Впорядкована D-матриця активності по горизонталі – це вичерпний перелік TF-векторів, призначених для перенесення всіх можливих комбінацій логічних дефектів у тестових наборах.

2. Впорядкована D-матриця активності за вертикаллю – це повний набір FT-векторів для дослідження комбінацій дефектів на всіх тестових наборах.

3. Упорядкована матриця активності є структурою розміром  $2^n \times 2^n$ , де рядки та стовпці впорядковані за стандартними двійковими адресами таблиць істинності, пов'язаних із тестами і несправностями. Ця структура використовується для вирішення завдань технічної діагностики. Координати матриці D-(X-, F-) визначаються двійковими адресами: вертикальна вісь відповідає таблицям істинності тестів, а горизонтальна – таблицям істинності несправностей, що перевіряються.

4. Матриця активності L є структурою розміром  $2^n \times 2^n$ , де кожний рядок формується або логічним вектором, або його інверсією.

5. Матриця відстаней являє собою H-структуру з розмірністю  $2^n \times 2^n$ . Кожний її рядок представлений координатами-відстанями, що визначаються між тестовими наборами та комбінаціями несправностей. Комбінації несправностей задані таблицями істинності, і виконується впорядкування матриці активності відповідно до правила переадресації координат:  $D=L_H$ .

6. Матриця X активних логічних входів має розмірність  $2^n \times 2^n$ , де кожна координата представлена n-розрядним вектором активності вхідних змінних. Активні змінні позначаються одиничними значеннями, тоді як пасивні або неперевірені входи виділяються крапками. Для побудови матриці

X одиничні координати матриці D замінюються векторними адресами таблиць несправностей, тоді як нульові координати підміняються крапками.

7. Матриця несправностей F (або карта тестування) слугує повною системою явних залежностей між вичерпним тестом і усіма можливими комбінаціями перевірених несправностей незалежно від складності логічної функції. Конкретне розташування координат на карті тестування обчислюється за формулою:

$$F_{ijt} = \bar{T}_{it} \leftarrow X_{ijt} = 1, \quad (2.8)$$

тобто 1-координати матриці X замінюються інверсними значеннями бітів вхідних тестових бінарних наборів з метою створення тестової карти F.

## 2.5 Тестування помилок 1-го та 2-го роду

Механізм створення карти тестування можна застосовувати для виявлення помилок першого (1-0) та другого (0-1) роду. При перевірці логічних функцій ці помилки є рівнозначними. Головне у цьому процесі — те, що комбінація помилок (несправностей) на тестовому наборі дає одиничну похідну за виходом логічного елемента:

$$T \oplus F = 1. \quad (2.10)$$

Логічна матриця містить помилки 1-го та 2-го роду, які характеризуються одиничними або нульовими рядками/стовпцями. Згодом ця інформація переноситься на матрицю дедукції та карту тестування, яка призначена для формування комбінацій зазначених помилок. Потім такі комбінації перевіряються за допомогою відповідних тестів.

Помилка 1-го роду – це умови на вхідних змінних, що спричиняють зміну стану від позитивного до негативного. У свою чергу, помилка 2-го

роду проявляється тоді, коли умови на вхідних змінних змінюють негативний стан на позитивний. Обидва типи помилок є значущими збоями у роботі вхідних змінних, які впливають на функціональність системи, змінюючи результативний стан на протилежний.

У логічному векторі 11100111 кількість значущих координат у форматі матриці (L, D, F), які визначають помилки 1-го і 2-го роду, аналізується і узагальнюється під час тестування для точнішого діагностування:

$$F = 2 \times |Y_0| \times |Y_1| = 2 \times 2 \times 6 = 24. \quad (2.11)$$

В матрицях обозначено (рис. 2.4): ошибки первого рода – зеленым цветом, второго рода – красным. Первый вариант строит карту тестирования логической функции методом перестановки активных координат L-матрицы по строкам, используя H-матрицу для получения матрицы дедукции. Второй вариант рассматривает перестановку активных координат L-матрицы по столбцам, используя H-матрицу, для получения матрицы дедукции.

L-matrix		H-matrix		D-matrix		F-matrix		Test\Fault	0	1
Y	1 1 1 0 0 1 1 1	0 1 2 3 4 5 6 7	T\F	000 001 010 011 100 101 110 111	T\F	000 001 010 011 100 101 110 111				
1	1 1	0 1 2 3 4 5 6 7	000	1 1	000	.11 1..				
1	1 1	1 0 3 2 5 4 7 6	001	1	001	.1	1.0			
1	1 1	2 3 0 1 6 7 4 5	010	1	010	.1	1.0			
0 1 1 1	1 1 1 1	3 2 1 0 7 6 5 4	011	1 1 1 1 1 1 1	011	.0 .0 .00 1..	1.0 10.			
0 1 1 1	1 1 1 1	4 5 6 7 0 1 2 3	100	1 1 1 1 1 1 1	100	.1 .1 .11 0..	0.1 01.			
1	1 1	5 4 7 6 1 0 3 2	101	1	101	.0	01.			
1	1 1	6 7 4 5 2 3 0 1	110	1	110	.0	0.1			
1	1 1	7 6 5 4 3 2 1 0	111	1 1	111	.00 0..				

а

L-matrix		H-matrix		D-matrix		T-matrix		T-matrix	
Y	1 1 1 0 0 1 1 1	0 1 2 3 4 5 6 7	F\T	000 001 010 011 100 101 110 111	F\T	000 001 010 011 100 101 110 111	000 001 010 011 100 101 110 111	000 001 010 011 100 101 110 111	
1	1 1	0 1 2 3 4 5 6 7	000		000				
1	1 1	1 0 3 2 5 4 7 6	001	1 1 1 1 1	001	.1 .0 .1 .0			
1	1 1	2 3 0 1 6 7 4 5	010	1	010	.1 .0 .1 .0			
0 1 1 1	1 1 1 1	3 2 1 0 7 6 5 4	011	1	011	.11 .0 .11	.00		
0 1 1 1	1 1 1 1	4 5 6 7 0 1 2 3	100	1	100	1.. 1.. 0..	0..		
1	1 1	5 4 7 6 1 0 3 2	101	1	101	1.0 1.0 0.1	0.1		
1	1 1	6 7 4 5 2 3 0 1	110	1 1 1 1 1	110	10. 10. 01. 01.			
1	1 1	7 6 5 4 3 2 1 0	111		111				

б

Рисунок 2.4 – Схема тестування помилок 1-го та 2-го роду

У першому випадку розглядаються дві таблиці істинності: одна використовується для ідентифікації тестів по рядках, інша — для визначення несправностей за стовпцями матриць (D, F).

У другому випадку все відбувається навпаки: тести ідентифікуються по стовпцях, а несправності — по рядках матриць (D, F). Очевидно, що результати для обох підходів побудови карти тестування є повністю однаковими, якщо аналізувати всі можливі комбінації несправностей. Єдина відмінність полягає у симетрії відносно головної діагоналі матриці: рядки та стовпці просто змінюються місцями, а координати, які були важливими у рядках, стають ключовими для стовпців, і навпаки.

## 2.6 Висновки до розділу 2

Виконано аналіз моделей, методів, метрик, технологій векторно-логічного комп'ютингу для подальшої розробки системи діагностування несправностей за картою тестування.

Таким чином, розглянуто методика нетрадиційного тестування логіки, яка базується на використанні логічної константи, представлені у формі матриці несправностей (рис. 2.5).

Ця матриця взаємодіє з логічною матрицею для отримання тестової карти у вигляді співвідношення вичерпного тесту та всіх можливих комбінацій перевірених несправностей. Такий підхід дозволяє розв'язувати завдання моделювання несправностей логічних функцій на основі вичерпного тесту, уникаючи використання алгоритмів моделювання.



### 3 СИСТЕМА ДІАГНОСТУВАННЯ НЕСПРАВНОСТЕЙ ЗА КАРТОЮ ТЕСТУВАННЯ

Розробляються процедури діагностування несправностей та синтезу тестів за картою тестування. Наведено пов'язані з ними алгоритми.

#### 3.1 Діагностування дефектів за картою тестування

Діагностика одиночних та кратних константних дефектів у логічній функціональності здійснюється шляхом виконання трьох паралельних операцій над рядками карти тестування. Використовуються дві основні процедури:

##### 1. Виявлення одиночних постійних несправностей

$$\mathbf{F}^S = \wedge \mathbf{T}_1 \wedge \overline{\vee \mathbf{T}_0}. \quad (3.1)$$

Теоретико-множинний еквівалент цієї формули:

$$\mathbf{F}^M = \cup \mathbf{T}_1 \setminus \cup \mathbf{T}_0. \quad (3.2)$$

##### 2. Діагностування кратних константних несправностей

$$\mathbf{F}^M = \vee \mathbf{T}_1 \wedge \overline{\vee \mathbf{T}_0}. \quad (3.3)$$

Теоретико-множинний еквівалент відповідно

$$\mathbf{F}^M = \cup \mathbf{T}_1 \setminus \cup \mathbf{T}_0. \quad (3.4)$$

Для проведення процедури діагностування потрібно виконати тестовий експеримент, у ході якого фіксуються тестові набори, що призвели до

неправильного функціонування. Такі набори позначаються одиницями, тоді як решта отримує нульові значення.

Діагностування здійснюється шляхом визначення сукупної множини несправностей, виявлених за допомогою одиничних тестових наборів карти тестування. Після цього ідентифікуються всі несправності, що відповідають нульовим тестовим наборам. Результатом віднімання другої множини від першої є фактична сукупність несправностей, присутніх у функціональності.

### 3.2 Процедура синтезу мінімального тесту за картою тестування

Для побудови мінімального тесту використовується карта тестування. Модель для формування мінімального тесту покриття одиничних логічних несправностей вхідних змінних елемента передбачає трансформацію карти тестування з подальшим виключенням несуттєвих тестових наборів, керуючись заданим критерієм ( $T_i \subseteq T_j$ ).

З метою виконання цього завдання необхідно провести перетворення карти тестування, виділивши стовпці, які відповідають перевірці одиничних несправностей вхідних змінних (рис. 3.1). На основі цих даних формується матриця, де кількість стовпців відповідає числу вхідних змінних.

### 3.3 Процедура визначення відношення включення

З матриці виключаються тестові набори, які дублюють властивості більш ефективних наборів. Для цього можна застосувати логічну схему (рис. 3.2), яка допомагає визначити, чи один вектор входить у склад іншого [8].

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	...1			.1.	.1.1				1.1	1.1.	1.11				111.	1111
0001	...0	..1.	..10			.11.	.110		1.0			11..	11.0			
0010			..01			.10.	.101	1...	1.1		1.01	11..	11.1			
0011		..0.				.10.	.100	1...	1.0	1.0.		11..	11.0			
0100		..1.	..11	.0..		.01.	.011	1...	1.1			10..				
0101		..1.	..10		.0.0	.01.	.010	1...	1.0				10.0			
0110		..0.	..01				.001	1...	1.1			10..	10.1		1001	
0111		..0.	..00				.00.	1...	1.0			10..	10.0	100.		
1000	...1	..1.	..11			.11.	.111		0.1			01..	01.1			
1001	...0			.1..	.1.0				0.0	0.1.	0.10				011.	0110
1010		..0.				.10.	.101	0...	0.1	0.0.		01..	01.1			
1011			..00			.10.	.100	0...	0.0		0.00	01..	01.0			
1100		..1.	..11		.0.1	.01.	.011	0...	0.1				00.1			
1101		..1.	..10	.0..		.01.	.010	0...	0.0			00..				
1110		..0.	..01			.00.		0...	0.1			00..	00.1	000.		
1111		..0.	..00				.000	0...	0.0			00..	00.0		0000	

а

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	...1			.1..												
0001	...0	..1.														
0010								1...								
0011		..0.						1...								
0100		..1.		.0..				1...								
0101		..1.						1...								
0110		..0.						1...								
0111		..0.						1...								
1000	...1	..1.														
1001	...0			.1..												
1010		..0.						0...								
1011								0...								
1100		..1.						0...								
1101		..1.		.0..				0...								
1110		..0.						0...								
1111		..0.						0...								

б

	0001	0010	0100	1000
0000	...1		.1..	
0001	...0	..1.		
0010				1...
0011		..0.		1...
0100		..1.	.0..	1...
0101		..1.		1...
0110		..0.		1...
0111		..0.		1...
1000	...1	..1.		
1001	...0		.1..	
1010		..0.		0...
1011				0...
1100		..1.		0...
1101		..1.	.0..	0...
1110		..0.		0...
1111		..0.		0...

в

Рисунок 3.1 – Схема синтезу мінімального тесту за матрицею тестування

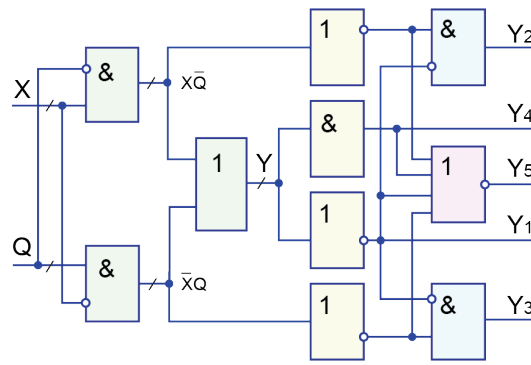


Рисунок 3.2 – Схема визначення відношення включення

При цьому застосовуються шість аналітичних виразів, реалізованих у логічній схемі для визначення відношень включення між двома логічними векторами:

$$Y = X\bar{Q}\bar{X}Q; \quad (3.5)$$

$$Y_1(X = Q) = [\vee Y] = [\vee(X\bar{Q}\bar{X}Q)]; \quad (3.6)$$

$$Y_2(X \subset Q) = [(\bar{X}Q)]; \quad (3.7)$$

$$Y_3(X \supset Q) = [(\bar{X}Q)]; \quad (3.8)$$

$$Y_4(X \cap Q = \emptyset) = [\wedge Y] = [\bar{Y}(X\bar{Q}\bar{X}Q)]; \quad (3.9)$$

$$Y_5(X \cap Q \neq \emptyset) = (Y_1 Y_2 Y_3 Y_4). \quad (3.10)$$

### 3.4 Процедура мінімізації рядків карти

Вихідні дані для процедури мінімізації рядків матриці – це матриця тестування логічної функціональності, з якої необхідно зробити матрицю тестування одиночних константних несправностей вхідних ліній (рис. 3.3) шляхом синтезу матриць M1, M2, M3.

M1	1				0				M2	1				0				M3	1				0			
	4	3	2	1	4	3	2	1		1	2	3	4	5	6	7	8		1	2	3	4	5	6	7	8
0000	1		1						0000	1	1						1	0000	1	1					1	
0001		1			1				0001	1			1				2	0001	1			1			2	
0010				1					0010									0010				1	1		3	
0011				1		1			0011			1		1			3	0100		1		1		1	4	
0100		1		1				1	0100		1		1			4	1000	1	1						5	
0101		1		1					0101									1001			1		1		6	
0110				1		1			0110									1010					1	1	7	
0111				1		1			0111									1100								
1000	1	1							1000	1	1					5	1101						1	1	8	
1001			1		1				1001			1		1		6	1110									
1010						1		1	1010					1		7	1111									
1011								1	1011																	
1100		1						1	1100																	
1101		1						1	1101	1					1	1	8									
1110						1		1	1110																	
1111						1		1	1111																	

Рисунок 3.3 – Схема перетворення карти тестування до компактного вигляду

### 3.5 Процедура покриття несправностей тестовим набором

Після виконання процедури мінімізації для цієї функціональності залишається лише вісім наборів матриці M3. Для них вирішується задача покриття всіх несправностей мінімальною кількістю тестових наборів. Це здійснюється шляхом формування термів кон'юнктивних нормальних форм. Вісім векторів утворюють відповідні диз'юнктивні терми, які записані у стовпцях мінімальної тестової матриці:

$$\begin{aligned}
 T &= (1v5)(2v4v5v8)(1v6)(3v4)(2v6)(3v7)(4v8)(7v8) = \\
 &= (1v5)(1v6)(3v4)(3v7)(4v8)(7v8)(2v6)(2v4v5v8) = \\
 &= (1v56)(3v47)(47v8)(2v62v64v65v68) = \\
 &= (1v56)(38v47)(2v64v65v68) = \\
 &= (138v147v3568v4567)(2v64v65v68) = \\
 &= 1238v13468v13568v1368v1247v1467v14567v14678 \cdot \\
 &23568v34568v3568v3568v24567v4567v4567v45678 = \\
 &= 1238v1368v1247v1467v3568v4567.
 \end{aligned} \tag{3.11}$$

Далі здійснюється розкриття диз'юнктивних термів для отримання множини кон'юнкцій, які охоплюють усі покриття несправностей тестовими наборами. Це включає й мінімальні терми, що забезпечують найменшу кількість тестових наборів, необхідних для повного покриття всіх одиночних несправностей:

$$T=1238v1368v1247v1467v3568v4567. \quad (3.12)$$

### 3.6 Алгоритм розв'язку задачі покриття

Алгоритм розв'язання задачі покриття для отримання мінімального тесту включає такі етапи:

1. Створюється кон'юнкція диз'юнктивних термів, складених за 1-координатами вибраних стовпців дедуктивної матриці.

2. Виконується логічне множення всіх диз'юнктивних термів у кон'юнктивній нормальній формі (КНФ) для отримання нормальної диз'юнктивної форми (ДНФ).

3. У результаті утворюється ДНФ, що охоплює всі можливі варіанти покриття обраних несправностей за допомогою різних комбінацій тестових наборів. Мінімальний кон'юнктивний терм у цій ДНФ визначатиме мінімальний тест.

Пошук мінімальних дефектів на заданому тесті здійснюється подібно до матриці МЗ. Визначається мінімальна кількість дефектів, що охоплюють тестові набори з неправильною реакцією виходів. Алгоритм розв'язання задачі покриття включає такі кроки:

1. Будується кон'юнкція диз'юнктивних термів для одиночних координат обраних рядків дедуктивної матриці.

2. Здійснюється логічне множення всіх диз'юнктивних термів КНФ для отримання нормальної диз'юнктивної форми.



На рис. 3.5 показано три структурні фрагменти формування знаків несправностей, що перевіряються в алфавіті  $\{0,1, \langle.\rangle\}$  на координатах карти тестування, відповідних 1-координатам дедуктивної матриці.

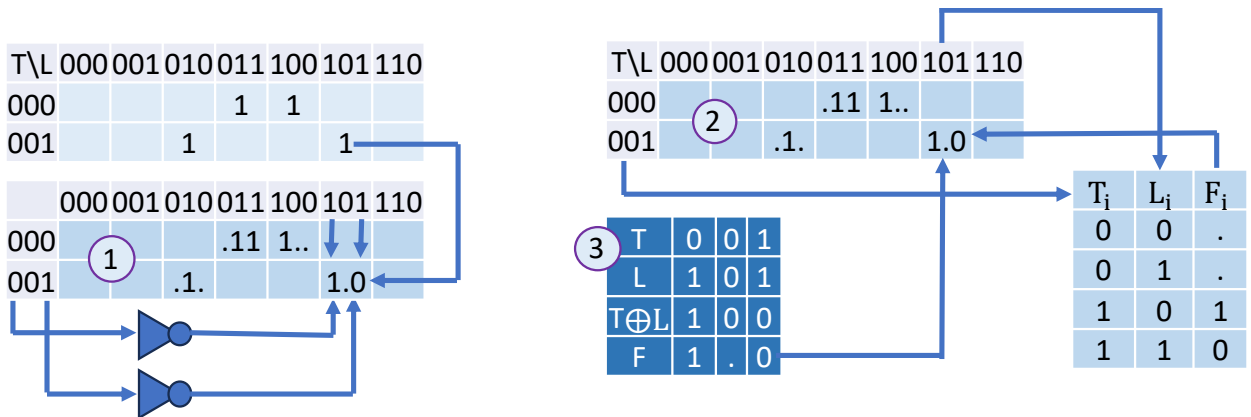


Рисунок 3.5 – Структурні фрагменти схем формування знаків несправностей: 1 – визначення 1-координат інверсними значеннями бітів двійкових тестових наборів; 2 – формування знаків несправностей F за таблицею істинності; 3 – визначення тестування функціональності декартовою хог-взаємодією адрес таблиці істинності за 1-координатами дедуктивної матриці

Перевірені несправності формуються лише за одиничними координатами адрес таблиці істинності L. Ці 1-координати завжди визначаються інверсними значеннями бітів двійкових тестових наборів (1). 0-координати адрес таблиці істинності в карті тестування завжди довизначаються точками, які означають відсутність несправностей, що перевіряються на цих вхідних змінних.

На рис. 3.5 також наведено таблицю істинності (2), яка формує знаки несправностей F на координатах карти тестування у функціональній залежності від змінних  $F=f(T,L)$ , що задаються адресами таблиць істинності.

Таким чином, карта тестування функціональності визначається (3) декартовою хог-взаємодією адрес таблиці істинності за 1-координатами дедуктивної матриці. Дві таблиці істинності – це компактні моделі

вичерпного тесту  $T$  та всіх логічних несправностей вхідних змінних  $L$ . Знаки перевірених несправностей формуються лише за одиничними координатами адрес таблиці істинності  $L$ . Нульові координати адрес  $L$  формують «.» точки, які означають відсутність несправностей, що перевіряються на вхідних змінних.

### 3.8 Процедура моделювання несправностей за таблицею істинності

Якщо задано дедуктивний вектор, то моделювання несправностей зводиться до трактування одиниць у таблиці істинності як несправностей, які перевіряються на визначеному двійковому наборі. У цьому випадку моделювання несправностей проходить через три етапи (рис. 3.6):

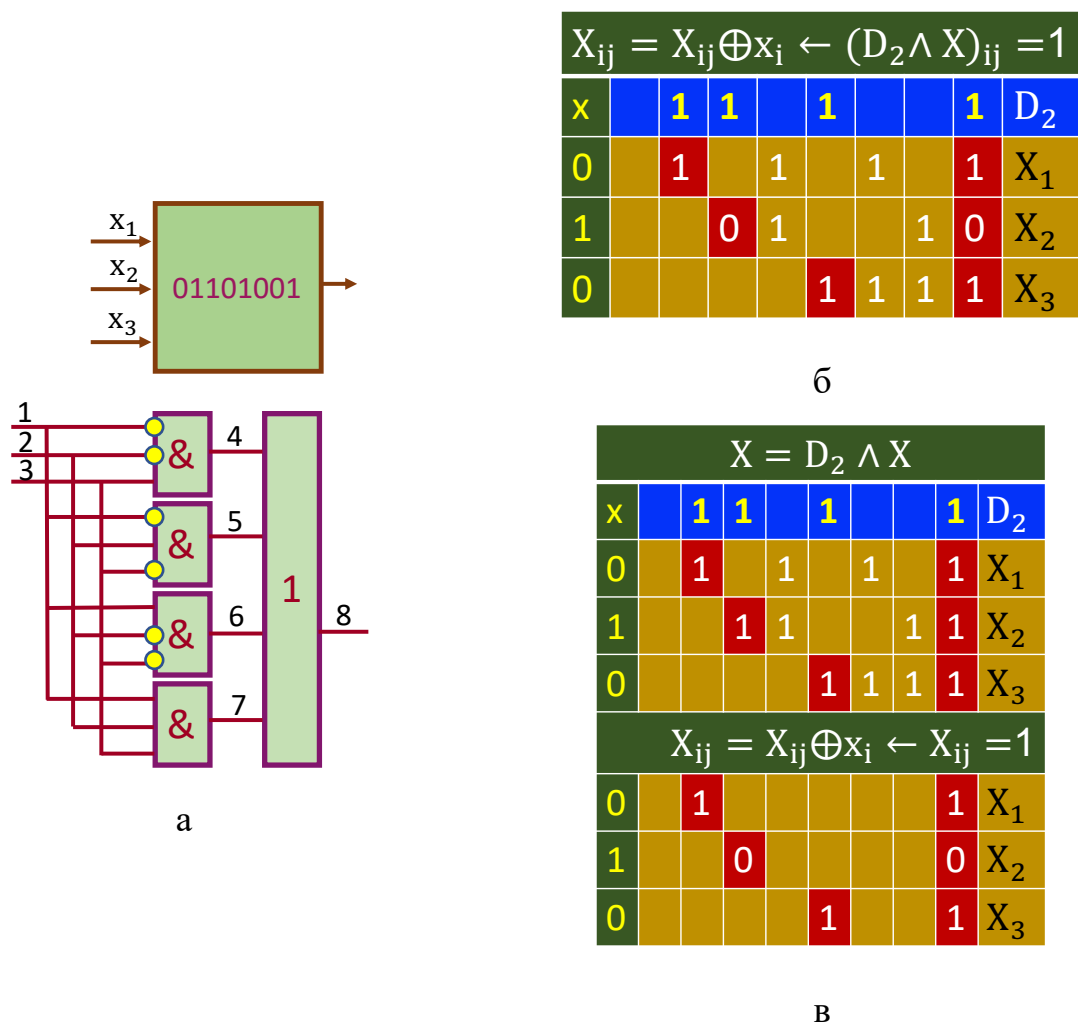


Рисунок 3.6 – Схеми моделювання несправностей за таблицею істинності

1. Активація координат із значенням "1" у таблиці істинності шляхом відповідного покриття їх одиничними значеннями дедуктивного вектора.
2. Визначення знаків несправностей для активованих координат на основі інверсних станів вхідних біт тестового набору.
3. Формування дедуктивного вектора.

### 3.9 Висновки до розділу 3

Розроблено компоненти системи діагностування несправностей за картою тестування, що містить процедури діагностування, синтезу тестів за картою тестування, мінімізації рядків карти, описано алгоритм розв'язку задачі покриття, моделювання несправностей за картою тестування.

## ВИСНОВКИ

1. Проаналізовано технологічні тенденції, що визначені компанією Gartner. Серед них напрямки, що спонукають удосконалення підходів до тестування, діагностування моделювання цифрових систем на кристалах – суперкомп'ютери на основі штучного інтелекту, автономні агенти, людиноподібні робочі роботи, цифрові двійники клієнта, просторові обчислення, що інтегрують фізичне та цифрове середовища, дозволяючи користувачам взаємодіяти з цифровим контентом у 3D-просторі.

2. Проаналізовано актуальні моделі, методи, технології векторно-логічного комп'ютингу, що застосовується у подальшому для розробки системи діагностування несправностей за картою тестування. Значний внесок до розвитку теорії векторно-логічного комп'ютингу зробили вчені кафедри АПОТ, що відзначено у роботах [4-14].

3. Розроблено компоненти системи діагностування несправностей за картою тестування, що містить процедури діагностування, синтезу тестів за картою тестування, мінімізації рядків карти, описано алгоритм розв'язку задачі покриття, моделювання несправностей за картою тестування.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Chandrasekaran A. 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>]
2. Akleman E. Computing Through Time: Metaverse / E. Akleman // *Computer*. – March 2025. – Vol. 58, no. 3. P. 8-8. doi: 10.1109/MC.2024.3522799.
3. Adel A. The Convergence of Intelligent Tutoring, Robotics, and IoT in Smart Education for the Transition from Industry 4.0 to 5.0 / A. Adel // *Smart Cities*. – 2024. – № 7. – P. 325-369. <https://doi.org/10.3390/smartcities7010014>
4. 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. doi: [10.15407/emodel.45.01.003](https://doi.org/10.15407/emodel.45.01.003)
5. 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. doi: [10.15407/emodel.45.02.016](https://doi.org/10.15407/emodel.45.02.016)
6. 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.
7. 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.
8. Vladimir Hahanov. Cyber-Physical Computing for IoT-driven Services. – New York: Springer, 2018. – 279 p. doi: 10.1007/978-3-319-54825-8.

9. 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. [doi: 10.1109/dessert58054.2022.10018769](https://doi.org/10.1109/dessert58054.2022.10018769)
10. 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. [doi: 10.15588/1607-3274-2023-2-5](https://doi.org/10.15588/1607-3274-2023-2-5)
11. Hahanova A. Vector-Deductive Faults-as-Address Simulation / A. Hahanova // International Journal of Computing. – 2023. – №22(3). – P. 328–334. [doi: 10.47839/ijc.22.3.3227](https://doi.org/10.47839/ijc.22.3.3227)
12. Векторні моделі логіки і структури для тестування та моделювання цифрових схем / Г.В. Хаханова, В.І. Хаханов, С.В. Чумаченко та ін. // Радіоелектроніка, інформатика, управління. – Запоріжжя: ЗНТУ. – 2021. – №3 . – С.69–85. [doi: 10.15588/1607-3274-2021-3-7](https://doi.org/10.15588/1607-3274-2021-3-7)
13. Vector Simulation of Logic Faults based on XOR-Relations / [V. Hahanov, S. Chumachenko, E. Litviniva et al.] // 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, Poland, 2021: proceedings. – IEEE, 2021. – P. 1041–1044. [doi: 10.1109/idaacs53288.2021.9660921](https://doi.org/10.1109/idaacs53288.2021.9660921)
14. Vector synthesis of fault testing map for logic / [V. Hahanov, W. Gharibi, S. Chumachenko et al.] // IAES International Journal of Robotics and Automation (IJRA). – 2024. – Vol. 13, № 3. – P. 293–306. [doi: 10.11591/ijra.v13i3.pp293-306](https://doi.org/10.11591/ijra.v13i3.pp293-306)