

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 другий (магістерський)
(рівень вищої освіти)

Метод векторного синтезу дедуктивної логіки для аналізу несправностей
комп'ютерних систем
(тема)

Виконав: студент II курсу, групи СКСм-21-1

Гайденко К.А.
(прізвище, ініціали)

Спеціальність
123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

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

Освітня програма
Спеціалізовані комп'ютерні системи
(повна назва освітньої програми)

Керівник проф. Чумаченко С.В.

Допускається до захисту

Зав. каф. АПОТ




(підпис)
2022 р.

Чумаченко С.В.
(прізвище, ініціали)

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Автоматизації проектування обчислювальної техніки
Рівень вищої освіти другий (магістерський)
Спеціальність 123 Комп'ютерна інженерія
(шифр і назва)
Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри АПОТ 
Чумаченко С.В.
(підпис)

« ___ » _____ 2022 р.

**ЗАВДАННЯ
НА КВАЛІФАЦІЙНУ РОБОТУ**

студентові Гайденко Кирилу Артемовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод векторного синтезу дедуктивної логіки для аналізу несправностей комп'ютерних систем

затверджена наказом по університету від 14 листопада 2022 р. № 1478 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 12 грудня 2022 р.

3. Вихідні дані до роботи _____

Моделі та методи аналізу несправностей

Матричні структури даних

Елементи булевої логіки

4. Перелік питань, що потрібно опрацювати у роботі _____

Аналіз предметної області

Аналіз сучасних технологічних тенденцій у проектуванні та діагностиці

Векторні структури даних

Векторно-дедуктивний метод

Алгоритм синтезу дедуктивних формул

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____
слайди презентації – 20 _____


6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)


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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів	Примітка
1	Отримання завдання	01.09.2022-05.09.2022	
2	Аналіз предметної області, сучасних технологічних тенденцій у design & test	07.09.2022-21.09.2022	
3	Метод векторного синтезу дедуктивної логіки	22.09.2022-05.10.2022	
4	Синтез дедуктивних формул транспортування несправностей	06.10.2022-19.10.2022	
5	Алгоритм синтезу дедуктивних векторів	20.10.2022-27.11.2022	
6	Оформлення пояснювальної записки	28.11.2022-30.11.2022	
7	Оформлення графічного матеріалу	01.12.2022-05.12.2022	
8	Перевірка виконаного проекту керівником	06.12.2022-09.12.2022	

Дата видачі завдання 01 вересня 2022 р.

Студент 
(підпис)

Керівник роботи  проф. Чумаченко С.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 61 с., 9 рис., 43 джерела.

ЦИФРОВИЙ ПРИСТРІЙ, НЕСПРАВНІСТЬ, МОДЕЛЬ, МЕТОД, АРХІТЕКТУРА, КІБЕРФІЗИЧНИЙ КОМП'ЮТИНГ, МЕТРИКА, ЦИФРОВА СХЕМА, ВЕКТОРНА ФОРМА ЛОГІКИ, ДЕДУКТИВНА МАТРИЦЯ, ТАБЛИЦЯ ІСТИННОСТІ, ДЕДУКЦІЯ, ДЕДУКТИВНО-ВЕКТОРНИЙ МЕТОД, ВЕКТОРНА МОДЕЛЬ ДЕФЕКТІВ.

У магістерській роботі розглядаються питання, пов'язані зі створенням моделей, методів аналізу несправностей комп'ютерних систем.

Мета дослідження – зменшення часу проектування дедуктивних формул для транспортування списків несправностей через логічні елементи та схеми RTL-рівня за рахунок використання паралельних регістрових процедур аналізу векторних структур даних.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- проаналізувати сучасні технологічні тенденції;
- виконати аналітичний огляд моделей та методів для аналізу несправностей;
- вдосконалити векторно-дедуктивний метод синтезу формул для транспортування списків несправностей;
- розробити алгоритм синтезу дедуктивних векторів.

Об'єкт дослідження – цифрові системи на кристалах.

Предмет дослідження – матричні структури, що використовуються для створення методів і алгоритмів паралельного розподілу та об'єднання векторів рядків і стовпців.

ABSTRACT

The explanatory note contains: 61 pages, 9 figures, 43 sources according to the list of links.

DIGITAL DEVICE, FAULT, MODEL, METHOD, ARCHITECTURE, CYBERPHYSICAL COMPUTING, METRICS, SYSTEM-ON-CHIP, VECTOR LOGIC FORM, DEDUCTION MATRIX, TRUTH TABLE, DEDUCTION, DEDUCTION-VECTOR METHOD, DEFECT VECTOR MODEL.

The master's work deals with issues related to the creation of models and methods for analyzing faults in computer systems.

The purpose of the investigation is to reduce the time for designing deductive formulas for transporting fault lists through logical elements and RTL level circuits by using parallel register procedures for analyzing vector data structures.

To achieve the goals set, it is necessary to solve the following tasks:

- analyze modern technological trends;
- perform an analytical review of models and methods for fault analysis;
- to improve the vector-deductive method of synthesis of formulas for the transportation of fault lists;
- develop an algorithm for the synthesis of deductive vectors.

The object of research is digital systems on crystals.

The subject of research is matrix structures used to create methods and algorithms for parallel distribution and combining of row and column vectors.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Сучасні технологічні тенденції у Desing & Test	10
1.2 Огляд моделей та методів	14
1.3 Ключові технологічні тренди 2023	21
1.4 Висновки до розділу 1	33
2 МЕТОД ВЕКТОРНОГО СИНТЕЗУ ДЕДУКТИВНОЇ ЛОГІКИ.....	34
2.1 Дедуктивне моделювання	34
2.2 Дедуктивно-векторний метод моделювання несправностей.....	37
2.4 Висновки до розділу 2	42
3 СИНТЕЗ ДЕДУКТИВНИХ ФОРМУЛ ТРАНСПОРТУВАННЯ НЕСПРАВНОСТЕЙ.....	43
3.1 Синтез дедуктивних функцій для AND-елемента	43
3.2 Синтез дедуктивних векторів для XOR-елемента	44
3.3 Синтез дедуктивних функцій для повторювача та інвертора.....	45
3.4 Висновки до розділу 3	46
4 АЛГОРИТМ СИНТЕЗУ ДЕДУКТИВНИХ ВЕКТОРІВ	47
4.1 Вихідні дані	47
4.2 Опис алгоритму.....	47
4.3 Порівняння результатів	51
4.4 Висновки до розділу 4	52
ВИСНОВКИ	53
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	56

ДОДАТОК А Графічний матеріал до кваліфікаційної роботи (презентація)	62
ДОДАТОК Б Тези доповіді, сертифікат	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

AI – штучний інтелект (Artificial Intelligent);

ALU – арифметико-логічний пристрій (Arythmetic Logic Unit);

BIST – вбудоване самотестування (Built-In Self-Test);

FPGA – програмована логічна інтегральна схема (Field Programmable Gate Array);

LUT – таблиця перекодування (Look-Up Table);

RTL – рівень регістрових передач (Register Tranfer Level);

SRAM – статична пам'ять з довільним доступом (Static Random Access Memory);

STAR – система самотестування та ремонту пам'яті;

QEC – квантова корекція помилок;

САПР – система автоматизована (Computer-Aided Design);

ПЛІС – програмована логічна інтегральна схема;

ШІ – штучний інтелект.

ВСТУП

У роботі розглядаються питання, пов'язані зі створенням моделей та методів аналізу несправностей комп'ютерних систем.

Мета дослідження – зменшення часу проектування дедуктивних формул для транспортування списків несправностей через логічні елементи та схеми RTL-рівня за рахунок використання паралельних регістрових процедур аналізу векторних структур даних.

Задачі дослідження орієнтовані на удосконалення та створення моделей, методів аналізу несправностей комп'ютерних систем:

- проаналізувати сучасні технологічні тенденції;
- виконати аналітичний огляд моделей та методів для аналізу несправностей;
- вдосконалити векторно-дедуктивний метод синтезу формул для транспортування списків несправностей;
- розробити алгоритм синтезу дедуктивних векторів.

Об'єкт дослідження – цифрові системи на кристалах.

Предмет дослідження – матричні структури, що використовуються для створення методів і алгоритмів паралельного розподілу та об'єднання векторів рядків і стовпців.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Наводиться огляд моделей, методів та технологічних рішень, орієнтованих на створення ефективних засобів, які надають можливість виконувати аналіз несправностей цифрових пристроїв.

1.1 Сучасні технологічні тенденції у Desing & Test

Мотивація дослідження продиктована декількома факторами: квантові застосування елементарних read-write транзакцій на memory-driven комп'ютері з векторним описом логіки, надмірність пам'яті для зберігання інтерпретативних гнучких моделей логіки, використання даних як адрес для підвищення швидкодії дедуктивного моделювання.

Квантові помилки [1, 2] є платою за ефективні квантові обчислення комбінаторних задач. Вони складаються не тільки з помилок перестановки бітів, що робить цей простий код із повторенням трьох кубітів непридатним для захисту від усіх можливих квантових помилок. описав Схеми кодування, описана Пітером Шором, для одного логічного кубіту у дев'ять фізичних кубітів шляхом вбудовування коду повторення в інший код. Надмірність Шора захищає від довільної квантової помилки на будь-якому фізичному кубіті. У середині квантового комп'ютера інформація має унікальні властивості, порівняно з даними в стандартній мікроелектроніці. Апаратне забезпечення квантового комп'ютера більш ніж мільярд трильйонів разів частіше виходить з ладу. Ця величезна сприйнятливність до помилок є єдиною найбільшою проблемою, що заважає квантовим обчисленням реалізувати свої заявлені можливості. Підхід, відомий як квантова корекція помилок (QEC) може вирішити цю проблему. Але ці експерименти ще не досягли рівня якості та складності, необхідного для зниження

загального рівня помилок у системі. Навіть у стійкій до відмов архітектурі, як у класичному, так і в квантовому комп'ютері, додаткова складність генерує нові шляхи до помилок. Поточні оцінки корекції або поріг безбитковості для квантового обладнання становить приблизно 1 помилку з 1000 операцій. Бажано б довести цей показник до рівня щонайменше 5 помилок на 1000 операцій.

Таким чином, надмірність завжди добре впливає на класичний та квантовий комп'ютинг для надійного безпомилкового функціонування. Векторна форма опису процесів та явищ прийнятлива як для квантового, так і класичного комп'ютера.

У [3-7] запропоновано ряд підходів до тестування (BIST), діагностування та відновлення працездатності структур пам'яті на основі надмірності, а також до побудови та оцінки алгоритмів аналізу надмірності на основі векторів переваг для запам'ятовуючих пристроїв із запасними елементами. Експерименти щодо застосування нових алгоритмів самотестування та ремонту (STAR) пам'яті SRAM типу показали ефективність запропонованого підходу. Простежується тенденція: на регулярній пам'яті для обробки великих даних усі завдання щодо design and test, diagnosis and repair вирішуються технологічно простіше, ніж на класичному комп'ютері, який використовує стандартний процесор на основі ALU.

Вузьким місцем класичного комп'ютера є канал взаємодії між повільною пам'яттю та швидкодіючим логічним процесором. В даний час продуктивність пам'яті збільшилася в десятки разів, тому канал зв'язку з логічним процесором стає непринятною перешкодою підвищення продуктивності комп'ютерної архітектури. Вихід спостерігається у перенесенні всіх обчислень на пам'ять, прибрати логічний процесор з архітектури комп'ютера, вирішивши заодно проблему вузьких місць взаємодії пам'яті та логічного процесора (Y. Zorian, Yerevan, 2007 IEEE EWDTs).

Таким чином, якщо перейти на memory-driven комп'ютер, де функцію логіки повинні виконувати елементи пам'яті із записаними в ній векторами, то такий комп'ютер максимально близько орієнтований за структурами даних на квантові

паралельні обчислення, але не на квантову логіку, що створює великі проблеми, а на транзакції read-write у квантовій пам'яті.

Ефективні алгоритми моделювання несправностей для комбінаційних схем [8-10] відомі вже кілька десятків років. Однак послідовне моделювання відмов, яке часто використовується у додатках для тестування та стійкості до відмов, залишається дуже трудомістким завданням, особливо для великих схем. Пропонується новий дедуктивний метод для моделювання помилок на рівні RTL і на системній моделі високорівневих діаграм прийняття рішень. Прискорення моделювання досягається з допомогою ефективних структур даних, реалізованих до виконання сукупності операцій у алгоритмі дедуктивного моделювання несправностей. Експерименти на еталонних схемах RTL показують, що з допомогою цього досягається скорочення часу роботи до двох порядків проти моделювання несправності лише на рівні вентилів. Відомий вчений Raimund Ubar виділяє дедуктивне моделювання, як найбільш витончений математичний апарат, який має гарні перспективи для обробки складних цифрових схем.

Таким чином, реалізація дедуктивного методу на векторних структурах даних, поміщених у пам'ять, не має аналогів за швидкістю обробки складних цифрових елементів за рахунок істотної надмірності пам'яті. Метод має перспективи бути впровадженим у цифрові та квантові схеми для моделювання несправностей, оцінки якості тестів, діагностування та усунення помилок, немає аналогів по швидкодії обробки складних цифрових елементів з допомогою істотної надмірності пам'яті.

Публікації [11-18] присвячені генерації тестів для комбінаційних та послідовних схем, розглядаються питання моделювання різних типів несправностей, які існують у сучасній мікроелектроніці. Наголос робиться на компактизацію та стиснення даних, тестів, несправностей та діагностичної інформації. У більшості програм цифрових логічних схем функція схеми або

специфікується $(0,1)$, або не специфікується (байдуже) для кожної вхідної умови. Однак підкреслюється, що є додатки, в яких будь-яка з підмножини функцій є прийнятним рішенням, навіть якщо неможливо уявити всі функції термінів виведення байдуже. І тут кажуть, що функція гнучка. У цій роботі пропонується процедура синтезу гнучких функцій з урахуванням функціональних блоків, званих блоками порівняння.

Таким чином, компактизація даних для технічної діагностики є дуже важливою властивістю поряд з гнучкістю даних, можливістю використовувати їх для опису різних типів несправностей сучасних цифрових пристроїв. Всі ці властивості укладаються у векторну форму подання логічних функцій та несправностей.

Тема викликає в області design and test певний інтерес з боку дослідників, які займаються тестуванням [27] і верифікацією цифрових систем на кристалі в умовах технології нанoeлектроніки, що зароджується. Субатомний рівень деталізації чіпів вимагає розробки нових підходів у галузі синтезу тестів, моделювання відмов та діагностики з метою створення тестованих та відмовостійких обчислювальних систем [1-7]. Це підтверджується численними публікаціями у сегменті технологій проектування та тестування. Ця тема в бібліотеці IEEE Xplore підтверджена 253 359 публікаціями, серед яких на напрям, пов'язаний з моделюванням несправностей, припадає близько 20% (47 227) опублікованих робіт. Крім того, відповідну оцінку двох згаданих напрямів досліджень було опубліковано Springer.

Імена дослідників, які зробили значний внесок у теорію та практику design, test and fault simulation: Y. Zorian, R. Ubar, J. Bergeron, Z. Navabi, A. Jerraya, D. Armstrong, P. Prinetto, J. Abraham, H. Fujiwara, T. Nishida, X. Wang, P. Mueller, A. Ivanov, Irith Pomeranz; Sudhakar M. Reddy, Jaan Raik; Artur Jutman, M. Jenihhin, N. Takahashi, Хана Кубатова, E.J. Marinissen, V.D Agrawal, J.P. Roth, A. Matrosova, P. Parkhomenko, A. Birger, Roy Kaushik.

Основні проблеми у галузі моделювання несправностей слід визначити так:

- 1) висока обчислювальна складність алгоритмів комбінаторного аналізу на реєстровому рівні опису моделі [5, 7-10];
- 2) складність алгоритмів моделювання та симуляції послідовних схем, пов'язана з непередбачуваною кількістю ітерацій [7-10];
- 3) значний обсяг структур даних для аналізу цифрових систем на кристалі, що негативно позначається на продуктивності методів моделювання несправностей та синтезу тестів [3-7];
- 4) складність аналізу та синтезу тестів для логіки великої розмірності та їх структурна складність, пов'язана з розгалуженнями, що сходяться [11-13];
- 5) паралельне вирішення завдань тестування та імітації цифрових пристроїв [3-6, 8-10].

1.2 Огляд моделей та методів

У технічній діагностиці використовуються три основні форми опису процесів та явищ: таблична, аналітична, графова [19-26]. При цьому матриця (таблиця) і вектор є дві форми опису моделей, що переходять один в одного. Вектор є компактним видом таблиці істинності у вигляді впорядкованої послідовності станів виходу, якщо вхідні компоненти-адреси впорядковані за зростанням [19, 21, 23, 25, 26]. Матриця, при необхідності, перетворюється в одновимірний вектор для зручності паралельної обробки даних реєстрової пам'яті. Природно, досить просто відновити таблицю чи матрицю з векторної форми опису процесу чи явища. Далі фігурує вектор, як компактна та технологічна форма опису об'єктів, функцій та структур для *memory-driven computing*.

Дедуктивне моделювання, запропоноване понад 50 років тому Армстронгом [27], а також удосконалене багатьма авторами, включаючи [20, 22, 28, 29], досі є

найефективнішим засобом аналізу якості тестів та синтезу таблиць для пошуку дефектів.

У [30, 31] пропонується покращення процесу проектування за рахунок запровадження трьох рівнів ієрархії проекту та надмірності. Це дозволяє зменшити час проектування time-to-market та підвищити якість проекту yield.

Запропоновано методику [32, 33, 35, 36, 38, 39] підвищення якості цифрових проектів на основі технології самоперевірених та самотестованих схем за рахунок введення надлишкових елементів. Ідея надмірності буде використана у даній магістерській роботі.

У роботах [34, 37, 40] розвивається тема якості цифрових проектів з урахуванням надмірності, що становить трохи більше 25% від функціональності, з допомогою якої вирішуються проблеми самотестування цифрових і релейно-контактних схем, і навіть стійкого функціонування схем у разі несправностей.

Статтю [31] присвячено проблемі вдосконалення FPGA-компонентів, що розробляються для систем критичного застосування. FPGA-компоненти поліпшуються в контролепридатності їх схем і достовірності обчислюваних на них результатів для підтримки відмовостійких рішень, які є базовими в забезпеченні функціональної безпеки критичних систем. Відмовостійкі рішення потребують захисту від джерел кратних відмов, до яких відносяться приховані несправності. Вони можуть накопичуватися в значній кількості на протязі тривалого нормального режиму і порушувати функціональність відмовостійких схем з початком найбільш відповідального аварійного режиму. Захист від прихованих несправностей забезпечується контролепридатністю схем, яка націлена на прояв несправностей і тому повинна підтримуватися в комплексі з достовірністю результатів, беручи до уваги зниження достовірності при прояві несправностей. Завдання підвищення контролепридатності FPGA-компонента в нормальному режимі і достовірності результатів, що обчислюються в аварійному режимі, вирішується шляхом

використання природної версійної надмірності, властивої LUT-орієнтованій архітектурі. Ця надмірність проявляється в існуванні множини версій програмного коду, що зберігають функціональність FPGA-компонента при одній і тій же його апаратній реалізації. Контролепридатність FPGA-компонента і достовірність обчислюваних результатів розглядаються з урахуванням характерних несправностей LUT-орієнтованої архітектури. Ці несправності досліджені з позиції несуперечності їх прояву і маскуванню відповідно в нормальному і аварійному режимі на версіях програмного коду. Несправності ототожнюються зі спотворенням бітів в пам'яті LUT вузлів. Біти, що спостерігаються тільки в аварійному режимі, є потенційно небезпечними, оскільки можуть приховувати несправності в нормальному режимі. Переміщення потенційно небезпечних бітів на контролепридатні позиції, які спостерігаються в нормальному режимі, виконується шляхом вибору відповідних версій програмного коду і організації роботи FPGA-компонента на декількох версіях. Експерименти, що проведені з FPGA-компонентом на прикладі матричного помножувача двійкових кодів, показали ефективність використання природної версійної надмірності LUT-орієнтованої архітектури для вирішення проблеми прихованих несправностей.

Стаття [32] присвячена проблемі перевіреності схем як невід'ємного елемента забезпечення функціональної безпеки інформаційно-керуючих охоронних систем моніторингу об'єктів підвищеної небезпеки в енергетичній, транспортній, військовій, космічній та інших галузях промисловості з метою попередження аварій, та зменшити виникнення їх наслідків. Відзначено ключову роль перевіряльності у трансформації відмовостійких структур, що використовуються в таких системах, у відмовостійкі. Показано проблеми логічної перевіреності, у тому числі проблему прихованих несправностей, властивих системам безпеки при сучасному проектуванні її компонентів з використанням матричних структур. Було запропоновано доповнити логічну перевіряємість іншими формами, серед яких

найбільш перспективними є енергетично-орієнтована контрольованість, підтримана успішним розвитком зелених технологій у FPGA (Field Programmable Gate Array). Відзначено проблеми обмеженої точності оцінки та вимірювання температури, які проявились у розробці термічної випробувальності та термічних методів моніторингу схем. Нижня і верхня перевіряність ланцюгів по потужності визначається параметром споживання струму. Отримано аналітичні оцінки нижньої та верхньої перевіряльності схем за струмом споживання з урахуванням особливостей їх проектування на ПЛІС за допомогою сучасних САПР (Computer-Aided Design) на прикладі Quartus Prime Lite 18.1. Оцінено порогові значення струмів споживання в методах моніторингу ланцюгів для виявлення несправностей у ланцюгах загальних сигналів і коротких замикань в рамках нижньої та верхньої перевіряємості відповідно. Були проведені експерименти для оцінки нижньої та верхньої перевіряльності схем та порогових значень для представлених методів моніторингу на прикладі масштабованої схеми регістра зсуву, розробленої для FPGA. Показано залежності енергетичної нижньої та верхньої перевіряльності схем від зайнятості мікросхеми ПЛІС.

Стаття [33] присвячена аналізу проблем розвитку комп'ютерних систем в домені критичних додатків. Виділено основні тренди цього розвитку, які полягають в підвищених запитах до продуктивності на основі розпаралелювання обчислень, до обробки наближених даних і забезпечення функціональної безпеки відповідно до необхідності структурування під паралелізм і наближеність природного світу, а також до підвищення відповідальності щодо прийнятих рішень. Проведено аналіз проблем, що виникають при реалізації існуючих рішень. Відзначено відставання теорій, обмежених моделлю точних даних, від практики в обробці наближених даних для сучасних систем, які отримують вихідні дані від датчиків, включаючи системи критичного застосування. Розкрито проблеми матричних структур, які лежать в основі проектування сучасних комп'ютерних систем і демонструють

низьку ефективність у продуктивності та енергоспоживанні, а також у забезпеченні функціональної безпеки, важливої для критичних додатків. Відзначено застосування відмовостійких рішень як основи функціональної безпеки і недовіру до цих рішень, яке проявляється в практиці використання небезпечних імітаційних режимів. Вони відтворюють аварійні умови для підвищення контролепридатності у вирішенні проблеми прихованих несправностей, оскільки відмовостійке рішення не стає відмовобезпечним при дефіциті контролепридатності. Дан аналіз джерел розглянутих проблем і можливостей їх вирішення з позиції ресурсного підходу, який ідентифікує проблему прихованих несправностей як проблему зростання з відставанням компонентів від розвитку системи. Показана роль матричних структур у відставанні компонентів і необхідність вирішення проблеми шляхом розвитку версійної надмірності для цих структур. Запропоновано метод введення версійної надмірності в матричну структуру на основі сильно пов'язаних версій для вирішення проблем відмовостійкості та контролепридатності в комплексі. Ефективність методу оцінена на прикладі матричного помножувача з використанням його програмної моделі.

У [34] запропоновано нову структуру комбінаційного пристрою самоперевірки, де на основі властивостей парності та кодів Бергера, а також коду з виявленням усіх подвійних помилок в інформаційних векторах вирішується задача виявлення всіх одиночних несправностей логічних елементів. можна вирішити без трансформації структури вихідного пристрою. Розглянуто властивості двійкових кодів з виявленням усіх подвійних помилок, які можуть бути використані при побудові запропонованої структури. Наводиться приклад побудови нової конструкції.

Робота [35] присвячена проблемам оперативного тестування обчислювальних схем, що працюють з наближеними даними. Розробка онлайн-тестування проходила під дією моделі точних даних, яка визначила призначення та надійність

свого методу лише для випадку точних даних. Особливості наближених розрахунків роблять помилку в більшості випадків несуттєвою. У цьому випадку методи онлайнного тестування виявляють нову властивість, щоб відхилити надійні результати, виявляючи несуттєві помилки. Цей факт створює проблему низької надійності методів on-line тестування при перевірці результату. Показано шляхи підвищення надійності методів on-line тестування при перевірці результату з використанням можливостей наближених розрахунків. Запропоновано нові методи on-line тестування, засновані на цих способах. Розроблено методи перевірки залишків для усічених операцій з мантисами, методи використання природної інформаційної надлишковості результатів арифметики та методи оцінки розміру помилок.

Стаття [36] присвячена проблемі прихованих несправностей, пов'язаних з розвитком комп'ютерних систем у вигляді їх трансформації в контрольновимірювальні системи безпеки. Приховані несправності не дозволяють забезпечити функціональну безпеку шляхом побудови відмовостійкого цифрового компонента без урахування їх перевіреності. Підвищення перевіреності шляхом імітації аварії наражається на неприйнятний ризик переходу в аварійний режим. Запропоновано підходи до вирішення проблеми в оперативному режимі з використанням методів підвищення перевіреності в штатному режимі та з аналізом як нормального, так і аварійного режимів.

У [37] розроблено нову техніку для побудови комбінаційних пристроїв із повністю самоконтрольними структурами, де виявляються будь-які одиничні несправності внутрішніх логічних елементів. Запропонована методика заснована на пошуку груп комбінаційних виходів пристроїв, в яких неможливі симетричні похибки (групи SI). При створенні таких груп розробник може вибрати варіанти реалізації пристрою самоперевірки, кожен з яких передбачає використання коду для

контролю відмов з виявленням будь-яких односпрямованих і асиметричних помилок (включно з ідентифікованою кратністю).

Робота [38] присвячена розробці об'єктів і методів онлайн-тестування. Вимоги, що пред'являються до комп'ютерних систем і оперативному тестуванню їх цифрових компонентів, аналізуються в ефективності використання ресурсів і в обмеженні прихованих процесів. Показано низький рівень виконання цих вимог у сучасних комп'ютерних системах і в схемах on-line тестування через домінування масивних структур, що відносяться до нижнього рівня розробки ресурсу. Запропоновано скорочення масивних структур шляхом перетворення сучасних комп'ютерних систем на багатопотокові системи порозрядних конвеєрів, що підвищують рівень у виконанні вимог. Розглянуто методи їх оперативного тестування з оцінкою значення результату та його похибки.

Стаття [39] присвячена розробці компонентів FPGA (Field Programmable Gate Array) для систем, пов'язаних із безпекою, як важливого напрямку підвищення функціональної безпеки об'єктів підвищеної небезпеки та самих систем керування з метою протидії аварійним та їх наслідки. Критичне застосування комп'ютерної системи диверсифікує її режим роботи на нормальний і аварійний, а також підвищує вимоги до відмовостійкості схем як основи функціональної безпеки. Відмовостійкі рішення не стають відмовостійкими в умовах недостатньої перевіреності, яка притаманна сучасним системам безпеки і проявляється в проблемі прихованих несправностей. Вони можуть накопичуватися під час нормального режиму та усувати відмовостійкість у аварійному режимі. Проекти FPGA з LUT-орієнтованою (Look-Up Table) архітектурою успадковують цю проблему в пам'яті LUT, яка використовується лише в аварійному режимі. Запропонований метод розвиває перевіряємість компонентів ПЛІС шляхом використання версійної надлишковості їх програмного коду. Періодична зміна версії програмного коду в звичайному режимі дозволяє адресувати пам'ять, яка

раніше використовувалася тільки з переходом в аварійний режим. Усі версії підтримують функціональність компонента FPGA, зберігаючи його апаратну реалізацію. Метод оцінює керованість і спостережливість пам'яті LUT і визначає версії, що підвищують її перевіряємість.

У [40] запропоновано нові відмовостійкі архітектури, які на сайті відомих архітектур подвійного та потрійного модульного резервування включають лише одну копію вихідної схеми. У нових структурах для вибору функцій, які потрібно виправити, використовується схема виявлення помилок сигналу. Схема побудована на основі методу булевого доповнення з перевіркою на парність обчислень. Представлено загальну архітектуру з корекційним сигналом на основі булевого доповнення. Ця архітектура дозволяє проектувати найпростіші відмовостійкі схеми. Наведено алгоритми побудови схеми помилок сигналу, а також приклади їх застосування.

Далі пропонується реалізація дедуктивного моделювання на основі векторної форми опису надмірної логіки транспортування несправностей [23-26], яка дає можливість суттєво спростити алгоритми моделювання з метою обробки цифрових схем великої розмірності.

1.3 Ключові технологічні тренди 2023

Компанія Gartner визначила 10 головних стратегічних технологічних тенденцій на 2023 рік [42]:

1. Цифрова імунна система
2. Прикладна спостережуваність.
3. AI Trust, управління ризиками та безпекою (AI TRISM).
4. Галузеві хмарні платформи.
5. Інжиніринг платформи.
6. Реалізація бездротового зв'язку.

7. Супердодатки.
8. Адаптивний штучний інтелект.
9. Метавсесвіт.
10. Сталі технології.

Як ці стратегічні технологічні тренди Gartner пов'язані з бізнесом? Можливо, бізнесу необхідно скоротити витрати, підвищити маржу чи реінвестувати. Або, можливо, підприємство все ще намагається зростати. Можливо, настав час зробити поворот — наново винайти бізнес-модель. Деяким із підприємств може навіть знадобитися зробити все це одночасно.

Бізнес-лідери та технологи можуть використовувати цей список для оцінки потенційного впливу цих технологічних тенденцій на конкретні стратегії їхніх підприємств, такі як зростання доходів, прискорення цифрових технологій, максимізація цінності даних або захист та розвиток бренду.

Ці тенденції можуть становити ризик або можливість для організації, і цей список допоможе створити технологічну дорожню карту, щоб вплинути на низку стратегічних цілей.

Дивлячись на те, коли ці тенденції стануть найбільш актуальними, можна встановити свій власний шлях (roadmap), розуміючи, що також не потрібно робити все одразу (рис. 1.1).



Рисунок 1.1 – Топ стратегічних технологічних трендів 2023 [42]

Які бізнес-пріоритети допоможуть вирішити технологічні тренди 2023?

Очікується, що тенденції 2023 року вплинуть на корпоративні стратегії протягом найближчих трьох років, дозволивши організаціям вирішити чотири ключові завдання:

- оптимізація стійкості, операцій чи довіри;
- масштабування вертикальних рішень, доставка продукції у будь-яку точку;
- новаторство взаємодії з клієнтами, прискорені відповіді чи можливості;

– пошук сталих технологічних рішень.

Тема 1: Оптимізація. Ці тенденції оптимізують ІТ-системи для підвищення надійності, покращують процес прийняття рішень на основі даних та підтримують цінність та цілісність систем штучного інтелекту (ІІ) у виробництві.

Цифрова імунна система створює покращену якість обслуговування клієнтів, комбінуючи кілька стратегій розробки програмного забезпечення для захисту від ризиків. Завдяки спостережуваності, автоматизації, а також екстремальному проектуванню та тестуванню вона забезпечує відмовостійкі системи, що знижують операційні ризики та ризики безпеки.

Цифрова імунна система (DIS) поєднує методи та технології проектування, розробки, експлуатації та аналітики програмного забезпечення для зниження бізнес-ризиків.

Надійна цифрова імунна система захищає програми та служби від аномалій, таких як наслідки програмних помилок або проблем з безпекою, роблячи програми більш стійкими, щоб вони могли швидко відновлюватись після збоїв. Це може знизити ризики для безперервності бізнесу, що виникають, коли критично важливі програми та служби зазнають серйозної загрози або взагалі перестають працювати.

«Підприємства стикаються з безпрецедентними труднощами при забезпеченні стійкості до відмови операційних середовищ, прискореної цифрової доставки та надійної взаємодії з кінцевими користувачами, — говорить Йоакім Хершманн, старший директор-аналітик Gartner. «Бізнес очікує, що він матиме можливість швидко реагувати на зміни ринку та швидко впроваджувати інновації. Кінцеві користувачі очікують більшого, ніж якісна функціональність — вони хочуть високої продуктивності, а їх транзакції та дані мають бути безпечними та задовільними під час взаємодії».

Чому цифрова імунна система має значення?

Цифрова імунна система поєднує в собі ряд методів і технологій від проектування програмного забезпечення, розробки, автоматизації, експлуатації та аналітики, щоб забезпечити чудовий інтерфейс користувача (UX) і зменшити системні збої, що впливають на ефективність бізнесу. DIS захищає програми та служби, щоб зробити їх більш стійкими до відмов і швидко відновлюватися після збоїв.

Під час недавнього опитування Gartner про подолання бар'єрів на шляху до цифрового виконання майже половина респондентів (48%) заявили, що основною метою їх цифрових інвестицій є покращення якості обслуговування клієнтів (CX). DIS буде мати вирішальне значення для забезпечення того, щоб CX не наражався на ризик через дефекти, збої системи або аномалій, таких як програмні помилки або проблеми з безпекою.

Gartner очікує, що до 2025 року організації, які інвестують у створення цифрового імунітету, збільшать задоволеність клієнтів за рахунок скорочення часу простою на 80%.

Прикладна спостережуваність працює на основі даних, наданих організацією, використовуючи штучний інтелект для аналізу та вироблення рекомендацій, які дозволяють підприємству приймати більш швидкі та точні рішення у майбутньому. При систематичному застосуванні це може зменшити затримку відповіді та оптимізувати бізнес-операції у режимі реального часу.

AI TriSM підтримує модель управління на основі штучного інтелекту, надійність, справедливість, ефективність та захист даних. Він поєднує в собі методи пояснення результатів штучного інтелекту, швидкого розгортання нових моделей, активного управління безпекою штучного інтелекту та контролю конфіденційності та етики.

Тема 2: Масштабування. Ці технологічні тенденції прискорюють вертикальні пропозиції, збільшують швидкість доставки продукції та забезпечують можливість підключення всюди.

Галузеві хмарні платформи, що поєднують SaaS, PaaS та IaaS з адаптованими галузевими функціями, які організації можуть використовувати, щоб легше адаптуватися до безперервного потоку збоїв у своїй галузі.

Інжиніринг платформи надає перевірений набір інструментів, можливостей та процесів, упакованих для зручності використання розробниками та кінцевими користувачами. Це підвищить продуктивність кінцевих користувачів та знизить навантаження на команди розробників.

Реалізація сервісів бездротового зв'язку охоплює надання послуг бездротової мережі звідусіль, включаючи традиційні обчислення кінцевих користувачів, підтримку периферійних пристроїв, рішення для цифрового маркування. Такі мережі виходять далеко за межі простого підключення, надаючи інформацію про місцезнаходження та іншу інформацію в реальному часі, а також аналітичні дані, та дозволяють системам безпосередньо збирати мережну енергію. Gartner використовує термін «реалізація цінності бездротового зв'язку» для опису низки способів, за допомогою яких бездротові технології наступного покоління незабаром забезпечать нові та покращені послуги та зменшать капітальні витрати.

Тема 3: Новаторство. Ці тенденції зосереджені на забезпеченні можливості зміни бізнес-моделі, оновлення взаємодії зі співробітниками та клієнтами та прискоренні стратегій для виходу на нові віртуальні ринки.

Супердодатки являють собою щось більше, ніж складові програми, що об'єднують послуги. Супердодаток поєднує функції програми, платформи та екосистеми в одному додатку, надаючи платформу третім сторонам для розробки та публікації власних міні-додатків.

Підприємствам слід звернути увагу: супердодатки можуть об'єднувати та замінювати кілька додатків для використання клієнтами чи працівниками та підтримувати зручну бізнес-екосистему.

Супердодаток схожий на швейцарський армійський ніж – із набором компонентів (міні-додатків), якими користувач може користуватися та видаляти за потреби. Супердодатки в тренді, тому що користувачі, особливо молоде покоління, народжене в епоху смартфонів, вимагають мобільного досвіду, який є потужним і простим у використанні.

«Ми також бачимо, що супердодатки викликають інтерес з боку перспективних організацій», – каже Джейсон Вонг, видатний віце-президент-аналітик Gartner. «Ці організації використовують стратегії компонування додатків і архітектури, щоб створити нові можливості для цифрового бізнесу у своїх галузях або на суміжних ринках».

Gartner очікує, що до 2027 року понад 50% населення планети будуть щодня активними користувачами кількох супердодатків. Концепція супердодатка також буде розширена, щоб включити корпоративні мобільні та настільні можливості, такі як робочий процес, спільна робота та платформи обміну повідомленнями.

Що пропонують супердодатки? Суперпрограма – це програма, яка надає кінцевим користувачам (клієнтам, партнерам або співробітникам) набір основних функцій, а також доступ до самостійно створених міні-програм. Супердодаток розглядається як платформа для створення екосистеми міні-додатків, яку користувачі можуть вибрати для активації для узгодженого та персоналізованого досвіду використання додатків.

Немає окремого магазину програм або ринку для міні-програм. Їх виявляють і активують користувачі суперпрограми, а після використання їх також можна легко видалити з інтерфейсу користувача.

Супердодатки працюють на платформі, яка надає численні, часто використовувані служби додатків, такі як обмін повідомленнями та оплата. Користувачі супердодатків можуть створювати персоналізовані користувацькі взаємодії (UX), вибираючи та встановлюючи вибрані ними міні-додатки (зосереджені на виконанні одного завдання). Супердодатки з часом розширяться, щоб підтримувати чат-ботів, технології Інтернету речей (IoT) і захоплюючий досвід, як-от метавсесвіт.

Як працюють супердодатки? Користувачі отримують доступ до ряду окремих сервісів через екосистему, для якої внутрішні групи розробників і зовнішні партнери створюють і розгортають модульні мікропрограми в супердодатку. Ця екосистема постачальників також підвищує цінність супердодатка, надаючи зручний доступ до ширшого спектру послуг у додатку.

Кілька постачальників технологій уже надають інструменти та платформи, які допомагають лідерам розробки програмного забезпечення створювати супердодатки.

Приклади: постачальники платформи як послуги (PaaS) пропонують рішення для хмарної платформи; фронтальні фреймворки, що дозволяють розгорнути міні-програми в веб-програмах і мобільних програмах; платформи розробки Multiexperience (MX); платформи прикладних програм із низьким кодом (LCAP); постачальники послуг розробки.

Користувачі можуть налаштувати свою суперпрограму, вибравши міні-програму, яку вони хочуть використовувати, коли вона їм потрібна, у суперпрограмі. Ключовим є обмін даними та включає просту автентифікацію користувача, таку як єдиний вхід (SSO) і відстеження налаштувань користувача або використання програми.

Адаптивний штучний інтелект дозволяє змінювати поведінку моделі після розгортання, використовуючи зворотний зв'язок у реальному часі, безперервно

перенавчати моделі та навчатися у середовищах виконання та розробки на основі нових даних та скоригованих цілей, щоб швидко адаптуватися до змінних реальних умов.

Адаптивний штучний інтелект поглинає знання навіть у процесі створення. Адаптивний штучний інтелект (ШІ), на відміну від традиційних систем ШІ, може переглядати власний код, щоб пристосуватися до змін у реальному світі, які не були відомі чи передбачені під час написання коду. Організації, які таким чином вбудовують адаптивність і стійкість до дизайну, можуть швидше й ефективніше реагувати на збої.

«Гнучкість і здатність до адаптації зараз є життєво важливими, як багато компаній зрозуміли під час недавніх криз у сфері охорони здоров'я та клімату», — каже видатний віце-президент-аналітик Gartner Ерік Бретену. «Адаптивні системи штучного інтелекту спрямовані на постійне перенавчання моделей або застосування інших механізмів для адаптації та навчання в середовищах виконання та розробки, роблячи їх більш адаптивними та стійкими до змін».

Gartner очікує, що до 2026 року підприємства, які прийняли інженерні методи штучного інтелекту для створення та керування адаптивними системами штучного інтелекту, перевершать своїх аналогів за кількістю та часом, необхідним для введення в дію моделей штучного інтелекту, принаймні на 25%.

Чому адаптивний ШІ важливий для бізнесу? Адаптивний штучний інтелект об'єднує набір методів (наприклад, розробка на основі агентів) і методів штучного інтелекту (тобто навчання з підкріпленням), щоб дозволити системам коригувати свої методи навчання та поведінку, щоб вони могли адаптуватися до мінливих обставин реального світу під час виробництва.

Вивчаючи моделі поведінки з минулого досвіду людей і машин, а також у середовищах виконання, адаптивний ШІ забезпечує швидші та кращі результати. Армія США та ВПС США, наприклад, створили систему навчання, яка адаптує свої

уроки до учня, використовуючи його індивідуальні сильні сторони. Він знає, чого навчати, коли перевіряти та як вимірювати прогрес. Програма діє як індивідуальний репетитор, підлаштовуючи навчання під учня.

А для будь-якого підприємства прийняття рішень є важливою, але дедалі складнішою діяльністю, яка вимагатиме від систем аналізу рішень більшої автономності. Але процеси прийняття рішень потребують реінжинірингу для використання адаптивного ШІ. Це може мати серйозні наслідки для існуючої архітектури процесів – і вимагає від зацікавлених сторін у бізнесі забезпечити етичне використання штучного інтелекту для відповідності вимогам і нормам.

Об'єднайте представників бізнесу, ІТ та відділів підтримки для впровадження адаптивних систем ШІ. Визначте випадки використання, надайте уявлення про технології та визначте джерела та вплив ресурсів. Як мінімум, бізнес-стейкхолдери повинні співпрацювати з даними та аналітикою, штучним інтелектом і розробкою програмного забезпечення для створення адаптивних систем ШІ. Розробка штучного інтелекту відіграватиме вирішальну роль у створенні та введенні в дію адаптивних архітектур ШІ.

Однак, зрештою, адаптивні системи забезпечать нові способи ведення бізнесу, відкриваючи двері для нових бізнес-моделей або продуктів, послуг і каналів, які зламують ізолюваність рішень.

Етапи впровадження адаптивного штучного інтелекту. Розробка штучного інтелекту забезпечує базові компоненти впровадження, введення в дію та управління змінами на рівні процесів, що забезпечує адаптивні системи ШІ. Але адаптивний штучний інтелект потребує значного посилення аспекту управління змінами в розробці штучного інтелекту. Якщо змінити лише кілька функцій, пов'язаних із цим принципом, це переможе.

Реінжиніринг систем для адаптивного штучного інтелекту суттєво вплине на співробітників, компанії та технологічних партнерів і не відбудеться миттєво.

По-перше, слід створити основи адаптивних систем штучного інтелекту, доповнивши поточні реалізації штучного інтелекту шаблонами проектування безперервного інтелекту та можливостями потоку подій – з часом переходячи до методів на основі агентів, щоб надати системним компонентам більше автономності.

Крім того, слід спростити для бізнес-користувачів впровадження штучного інтелекту та сприяйте управлінню адаптивними системами штучного інтелекту, додаючи чіткі та вимірні бізнес-індикатори через операційні системи, а також довіру до системи прийняття рішень.

Отже, адаптивний штучний інтелект забезпечує чудову та швидшу взаємодію з користувачем, адаптуючись до мінливих умов реального світу.

Розширення можливостей прийняття рішень і гнучкість відбуваються під час впровадження можливостей аналізу рішень.

ІТ-лідерам необхідно переробити різні процеси, щоб створити адаптивні системи штучного інтелекту, які можуть навчатися та змінювати свою поведінку залежно від обставин.

Метавсесвіт дозволяє людям відтворювати чи покращувати свою фізичну активність. Це може статися шляхом перенесення чи розширення фізичних дій у віртуальний світ чи шляхом перетворення фізичного світу. Це комбінаторна інновація, що складається з кількох технологічних тем та можливостей.

Тема 4: Стійкі технології. У 2023 році одних лише технологій буде недостатньо.

Стійка технологія є структурою рішень, що підвищує потужність та ефективність ІТ-послуг; забезпечує стійкість підприємства за допомогою таких технологій, як простежуваність, аналітика, програмне забезпечення для управління викидами та штучний інтелект; і допомагає клієнтам досягти своїх власних цілей у сфері сталого розвитку.

Стійкі технології не є однією ціллю; це структура цифрових рішень, яка забезпечує результати ESG.

Gartner очікує, що до 2025 року 50% ІТ-директорів матимуть показники ефективності, пов'язані зі стійкістю ІТ-організації. Але зосередження лише на сталості внутрішніх ІТ-операцій («стабільні ІТ») є занадто вузьким способом думати про стійкі технології. Замість цього також слід подумати про те, щоб забезпечити цілу низку стійких результатів за допомогою технологій.

Gartner визначає стійкі технології як структуру цифрових рішень, які можуть забезпечити екологічні, соціальні та управлінські (ESG) результати для підприємства та його клієнтів.

«Стійкі технології стають дедалі важливішими в операційному плані – наприклад, для оптимізації витрат, енергоефективності та використання активів – але вони також сприяють результатам ESG, таким як покращення самопочуття та забезпечення відстеження, необхідного для забезпечення відповідальної ділової практики», – каже головний аналітик Gartner Отум Станіш. «Стійкі технології також сприяють створенню нових бізнес-моделей і технологічних продуктів для кращого обслуговування клієнтів».

Інвестиції у стійкі технології також можуть підвищити операційну стійкість та фінансові показники, відкриваючи нові можливості для зростання.

Отже, можна зробити наступні висновки:

- 10 найкращих стратегічних технологічних тенденцій Gartner допоможуть задовольнити потреби вашого бізнесу в оптимізації, масштабуванні чи новаторстві;
- тенденції дозволяють привести технологічні інновації у відповідність до майбутніх стратегічних цілей підприємства;
- інвестиції у стійкі технології забезпечують операційні та фінансові вигоди та можуть створювати можливості для зростання.

1.4 Висновки до розділу 1

Таким чином, на підставі огляду літературних джерел, сучасних технологічних тенденцій у design & test, моделей і методів визначено основні проблеми у галузі моделювання несправностей. Дістало висновку, що реалізація дедуктивного моделювання на основі векторної форми опису надмірної логіки транспортування несправностей, надасть можливість суттєво спростити алгоритми моделювання для обробки цифрових схем великої розмірності.

Мета дослідження – зменшення часу проектування дедуктивних формул для транспортування списків несправностей через логічні елементи та схеми RTL-рівня за рахунок використання паралельних регістрових процедур аналізу векторних структур даних.

Задачі дослідження:

- 1) проаналізувати сучасні технологічні тенденції;
- 2) виконати аналітичний огляд моделей та методів для аналізу несправностей;
- 3) вдосконалити векторно-дедуктивний метод синтезу формул для транспортування списків несправностей;
- 4) розробити алгоритм синтезу дедуктивних векторів.

2 МЕТОД ВЕКТОРНОГО СИНТЕЗУ ДЕДУКТИВНОЇ ЛОГІКИ

Розглядається дедуктивне моделювання несправностей цифрових пристроїв. Удосконалено метод синтезу дедуктивних формул для схем великої розмірності, представлених RTL-елементами на основі опису елементів будь-якої складності у вигляді вектора вихідних станів комбінаційного пристрою.

2.1 Дедуктивне моделювання

Дедуктивне моделювання несправностей цифрових пристроїв є передовою технологією, яка обслуговує область проектування і тестування сучасних комп'ютерних систем. У цьому моделювання несправностей вирішує завдання оцінки якості тесту у класі одиночних константних дефектів.

Однак обчислювальна складність отримання дедуктивних формул, що оцінюється як n^3 , є досить складним завданням для функціональних схем RTL-рівня великої розмірності, тому дедуктивний метод, як правило, використовується тільки для цифрових схем, представлених на рівні вентилів.

Далі пропонується метод синтезу дедуктивних формул для схем великої розмірності, представлених RTL-елементами. Даний метод став можливим завдяки опису елементів будь-якої складності у вигляді вектора вихідних станів комбінаційного пристрою.

Основою вимірювання процесів та явищ у дискретному (двійковому) просторі є метрика, яка оперує трьома аксіомами (рефлексивність, симетричність та транзитивність) циклічної або замкнутої взаємодії між 1,2,3 компонентами.

Індукція (*inductio*) – логічне отримання висновку від частки до загального, де правильність гарантується достатньою чи вичерпною кількістю фактичних даних.

Дедукція (deductio) - логічне отримання висновку від загального до приватного, де правильність гарантується істинністю посилок-аксіом, що призводять до істинності наслідків-теорем.

На дедукцію працює аксіома конволюції (testing) простору процесів та явищ: «xor-відношення відстаней між кінцевим числом замкнених у циклі компонентів завжди дорівнює нулю» [19]:

$$\bigoplus_{i=1}^n d_i = 0.$$

Природно, що xor сума відстаней між замкнутими функціональними об'єктами, представленими у векторному вигляді, також дорівнює нулю.

Але істиною є і той факт, що xor-сума кубітних [19, 26] векторів функцій, як об'єктів

$$\text{and} = 0001, \text{or} = 0111, \text{xor} = 0110,$$

представлена на рис. 2.1, дорівнює нулю.

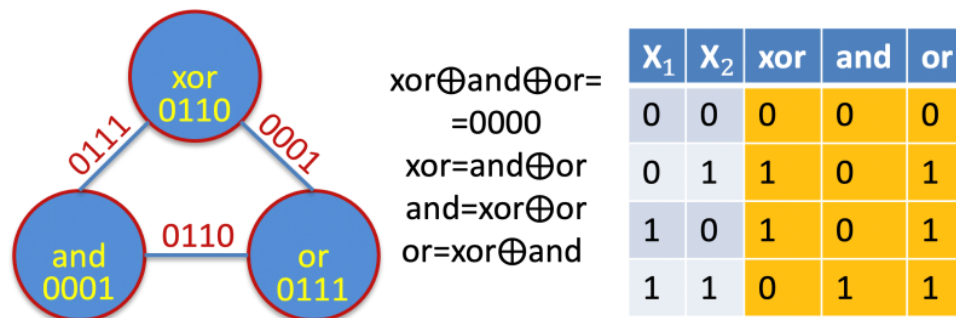


Рисунок 2.1 – Конволюція логічних функцій

Таким чином, виходить, що хог-відношення двох згаданих функцій між собою дорівнює третьої. У сукупності три функції (Xor, And, Or) за допомогою хог-відношення згортаються в нуль:

$$X \oplus A \oplus O = 0.$$

Математичне обґрунтування запропонованого методу ґрунтується на лемі про конволюцію трьох функцій від двох змінних [19]: існує тільки 4 пари логічних функцій, пов'язаних хог-відношенням, які в результаті дають хог-функцію:

1	2	3	4
$\oplus=0110$	$\oplus=0110$	$\oplus=0110$	$\oplus=0110$
$\wedge=0001$	$\bar{V}=1000$	$X_1=0011$	$\bar{X}_1=1100$
$v=0111$	$\bar{A}=1110$	$X_2=0101$	$\bar{X}_2=1010$

Тут перші два варіанти можна вважати парами логічних функцій, взаємодія яких дає хог-функцію. Інші дві є виродженими функціями від однієї змінної, які створюють хог-операцію. Якщо врахувати, що другий варіант є інверсія першого, можна зробити висновок у тому, що лише дві логічних функції (and, or) здатні створювати хог-вектор при хог-взаємодії.

Практична значимість лемі про конволюцію функцій полягає в наступному:

1) Оператор хог є унікальним та універсальним вимірювачем подібності-відмінності будь-яких кіберфізичних процесів та явищ.

2) Тільки дві логічні функції (and, or) своєю відмінністю створюють хог-операцію. Ще цікавіше те, що відмінність двох логічних функцій (and, or) одно їх відмінності. Виконання хог-операції над логічними функціями (and, or) і хог-операції.

3) Дві функції від однієї змінної (інверсія та повторення) також створюють xor-взаємодія.

4) Тріада логічних функцій (and, or, xor) формує векторну метрику для обчислення структурних:

$$S(a, b) = a_i \wedge_{i=1,n} b_i,$$

$$D(a, b) = a_i \oplus_{i=1,n} b_i,$$

і нормованих оцінок подібності-відмінності між процесами та явищами:

$$S_n(a, b) = \frac{\sum_{i=1}^n (a_i \wedge_{i=1,n} b_i)}{\sum_{i=1}^n (a_i \vee_{i=1,n} b_i)},$$

$$D_n(a, b) = \frac{\sum_{i=1}^n (a_i \oplus_{i=1,n} b_i)}{\sum_{i=1}^n (a_i \vee_{i=1,n} b_i)}.$$

5) Операція диз'юнкції (or) створює загальну метрику одиничних значень координат для виміру норми подібності-відмінності двох процесів чи явищ.

2.2 Дедуктивно-векторний метод моделювання несправностей

Моделювання складається з двох фаз:

- 1) Синтез дедуктивних формул на основі векторного опису RTL-логіки.
- 2) Моделювання несправностей логіки під час подання тестових впливів шляхом використання побудованих дедуктивних формул. Сама формула тут і далі

розглядається як похідна від векторної форми функціонального елемента, списків несправностей вхідних змінних та вхідного тестового впливу.

Для формування векторів, відповідних функцій вхідних змінних, необхідно синтезувати або мати таблицю істинності для функцій від змінних n .

На рис. 2.2 представлені таблиці для чотирьох, трьох, двох та однієї змінної, а також деякі функції-стовпці, для яких необхідно синтезувати дедуктивні вектори моделювання або транспортування несправностей через заданий функціональний елемент.

X_1	X_2	X_3	X_4	A	O	X	F		X_1	X_2	X_3	A	O	X	E		X_1	X_2	A	O	X		X_1	R	N
0	0	0	0	0	0	0	1		0	0	0	0	0	0	1		0	0	0	0	0		0	0	1
0	0	0	1	0	1	1	1		0	0	1	0	1	1	1		0	1	0	1	1		1	1	0
0	0	1	0	0	1	1	0		0	1	0	0	1	1	1		1	0	0	1	1				
0	0	1	1	0	1	0	0		0	1	1	0	1	0	0		1	1	1	1	0				
0	1	0	0	0	1	1	1		1	0	0	0	1	1	0										
0	1	0	1	0	1	0	1		1	0	1	0	1	0	0										
0	1	1	0	0	1	0	0		1	1	0	0	1	0	1										
0	1	1	1	0	1	1	0		1	1	1	1	1	1	1										
1	0	0	0	0	1	1	1																		
1	0	0	1	0	1	0	1																		
1	0	1	0	0	1	0	0																		
1	0	1	1	0	1	1	0																		
1	1	0	0	0	1	0	1																		
1	1	0	1	0	1	1	1																		
1	1	1	0	0	1	1	0																		
1	1	1	1	1	1	0	0																		

Рисунок 2.2 – Таблиці істинності для деяких логічних функцій

При синтезі дедуктивних векторів (формул) використовуються всі вектор-стовпці, що відповідають вхідним змінним X_1, X_2, X_3, X_4 , а також вектор-стовпець, який визначає логічну функцію, наприклад, А (and) або О (or), або Х (xor), або F,E (other functions).

Як приклад, можна розглядати функції-стовпці-змінні

$$X_1 = 0011, X_2 = 0101,$$

а також логічний вектор станів виходу $O=0111$ для побудови дедуктивного вектора.

Векторна модифікація дедуктивного методу моделювання несправностей. Далі розглядається трикутник хог-відношень для векторної модифікації дедуктивного методу моделювання несправностей, що використовує дедуктивні формули, синтез яких пропонується на рис. 2.3.

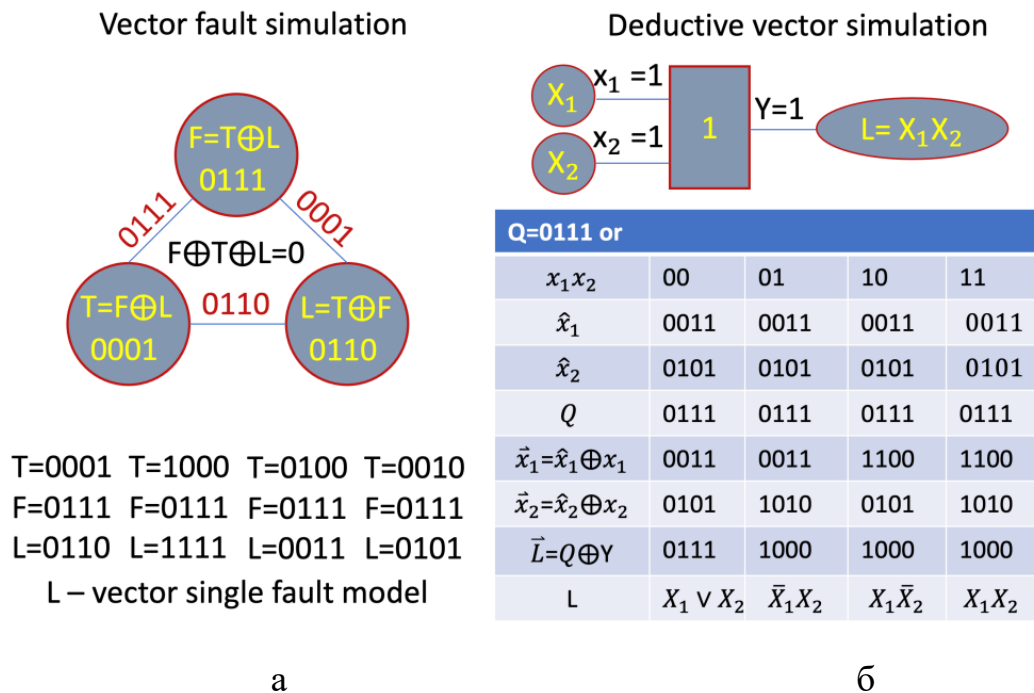


Рисунок 2.3 – Дедуктивно-векторне or-logic моделювання:

а – модель одиночних несправностей; б – процедури синтезу дедуктивних формул

На рис. 2.3,а вводиться нова модель одиночних несправностей, прив'язана до координат вектора, що представляє модель логічного об'єкта. При цьому список-вектор дефектів, що перевіряються, формується шляхом виконання хог-операції між функціональністю і тестовим набором, які також задаються у векторному вигляді.

На рис. 2.3,б представлені процедури синтезу дедуктивних формул для алгоритму транспортування дефектів на вихід логічного елемента, але при описі функціональностей та дефектів у векторному вигляді. Задаються функції змінних X_1, X_2 у векторному вигляді. Потім виконується інверсія координат векторів, якщо стан вхідної змінної X_i дорівнює одиниці. Виконується диз'юнкція (кон'юнкція) отриманих векторів, а далі до результуючого вектора застосовується процедура інверсії, якщо стан виходу логічного елемента дорівнює одиниці. Список-вектор L дефектів, що транспортуються на вихід, можна трансформувати до виду об'єднання (диз'юнкції), перетину (кон'юнкції) вхідних списків несправностей, представлених в останньому рядку.

Координати таблиці – двійкові вектори, що обробляються за такими правилами:

$$\bar{x}_1 = \hat{x}_1 \oplus x_1,$$

$$x_1 = \{0,1\} \rightarrow x_1 = 1,$$

$$\bar{x}_1 = 0011,$$

$$\bar{x}_1 = \hat{x}_1 \oplus x_1 = 1100,$$

$$x_1 = 0,$$

$$\vec{x}_1 = \hat{x}_1 \oplus x_1 = 0011,$$

$$\vec{x}_2 = \hat{x}_1 \oplus x_2 = 0,$$

$$\vec{x}_2 = \hat{x}_1 \oplus 0 = 0011 \oplus 0 = 0011,$$

$$\vec{x}_i = \hat{x}_1 \oplus x_i,$$

$$x_i = \{0,1\}.$$

Вводиться нова модель одиночних несправностей, прив'язана до координат векторів-стовпців, що представляють векторні компоненти моделі логічного елемента:

$$x_1 = 0011,$$

$$x_2 = 0101,$$

$$Y = 0111,$$

отримані зі стовпців наступної таблиці істинності:

x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

2.4 Висновки до розділу 2

Розглянуто дедуктивне моделювання несправностей цифрових пристроїв.

Удосконалено метод синтезу дедуктивних формул для схем великої розмірності, представлених RTL-елементами, на підставі опису елементів будь-якої складності у вигляді вектора вихідних станів комбінаційного пристрою.

Запропоновано та описано векторна модифікація дедуктивно-векторного методу моделювання несправностей, що використовує дедуктивні формули, із застосуванням трикутника хог-відношень.

3 СИНТЕЗ ДЕДУКТИВНИХ ФОРМУЛ ТРАНСПОРТУВАННЯ НЕСПРАВНОСТЕЙ

На основі різних функцій та однакових тестів будуються різні дедуктивні формули для транспортування несправностей. Ці моделі дозволяють протранспортувати всі входні несправності від зовнішніх входів до зовнішніх виходів.

3.1 Синтез дедуктивних функцій для AND-елемента

Синтез дедуктивних функцій транспортування несправностей для логічного елемента and на підставі $L = T \oplus F$ подано на рис. 3.1.

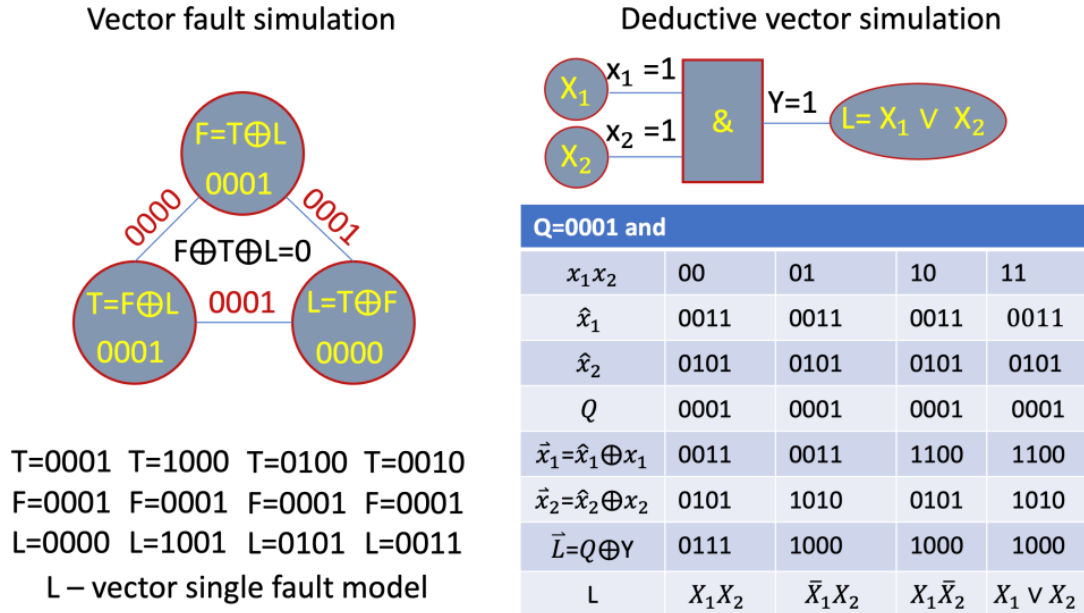


Рисунок 3.1 – Дедуктивно-векторне and-logic моделювання

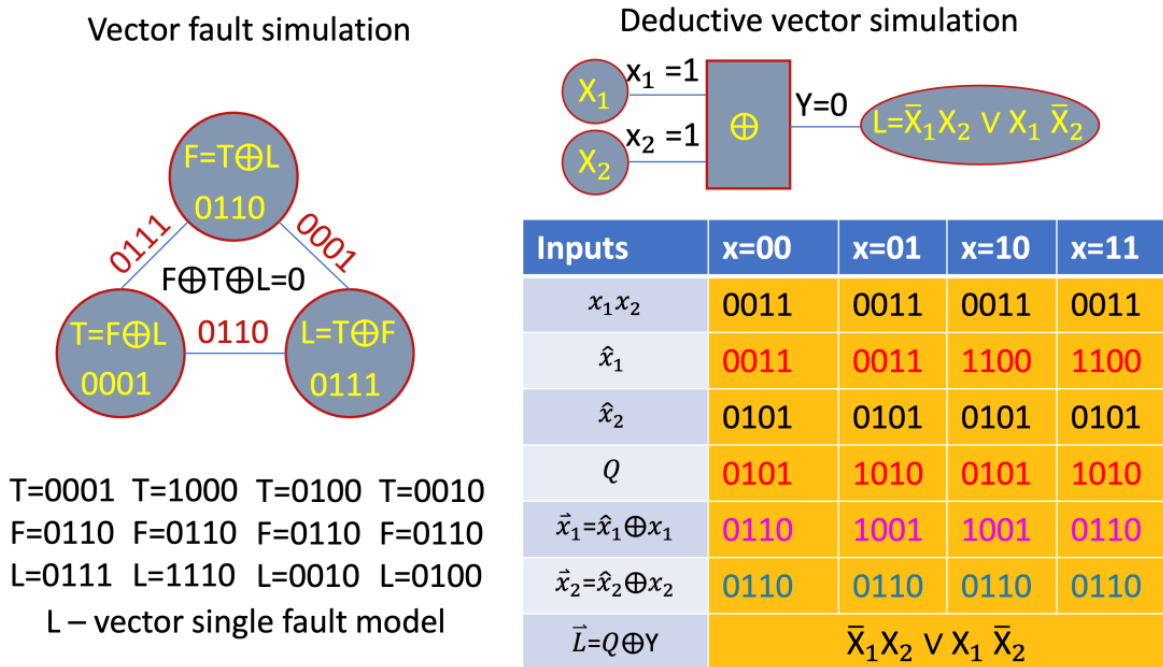
Ці моделі дозволяють протранспортувати всі вхідні несправності від зовнішніх входів до зовнішніх виходів.

3.2 Синтез дедуктивних векторів для XOR-елемента

Синтез дедуктивних векторів (формул) для транспортування списків вхідних несправностей через логічний елемент хор на підставі

$$L = T \oplus F$$

представлений на рис. 3.2,а.



а

б

Рисунок 3.2 – Дедуктивно-векторне хор-logic моделювання:

а – синтез дедуктивних векторів; б – синтез векторів несправностей

Цікавим є той факт, що аналітична формула

$$L = \bar{X}_1 X_2 \vee X_1 \bar{X}_2$$

транспортування списків вхідних дефектів на вихід логічного хог-елемента є однаковою для всіх тестових вхідних наборів.

На рис. 3.2,б подана процедура синтезу векторів несправностей L , які є залежними від тест-вектора T і функції-вектора F .

3.3 Синтез дедуктивних функцій для повторювача та інвертора

Синтез дедуктивних функцій для логіки повторювача та інвертора подано на рис. 3.3, де показані абсолютно однакові вирази транспортування списків несправностей на виходи елементів без будь-якої фільтрації.

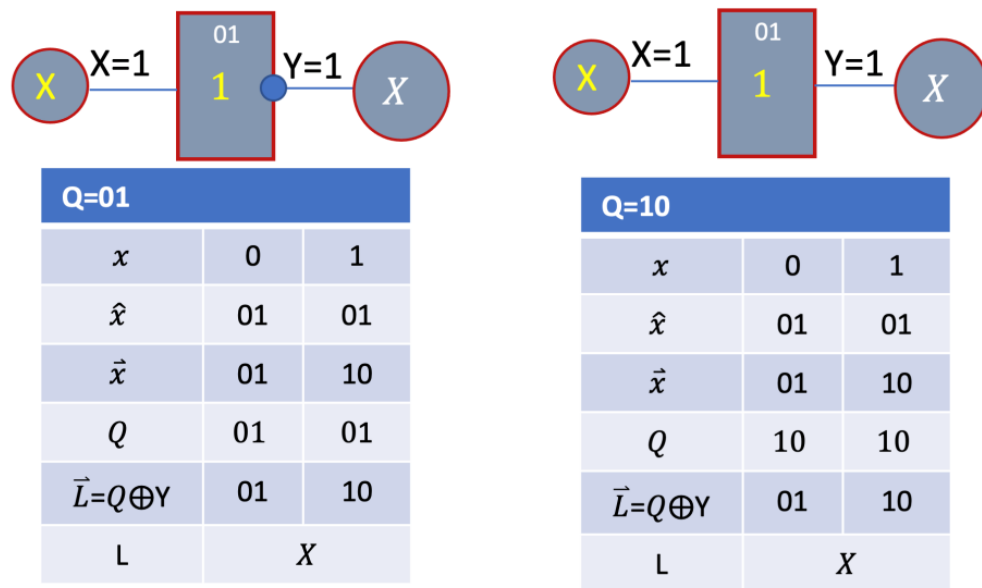


Рисунок 3.3 – Синтез дедуктивних функцій для повторювача та інвертора

Тут рядок $x \oplus \{0,1\}$ формує відношення інверсії вектора стану до одиничного значення вхідного сигналу, а рядок $Y \oplus \{0,1\}$ – інвертує вектор, якщо значення виходу елемента дорівнює одиниці.

3.4 Висновки до розділу 3

На основі різних функцій та однакових тестів побудовано різні дедуктивні формули для транспортування несправностей. Наведені моделі дозволяють протранспортувати всі вхідні несправності від зовнішніх входів до зовнішніх виходів.

4 АЛГОРИТМ СИНТЕЗУ ДЕДУКТИВНИХ ВЕКТОРІВ

Розглядається алгоритм синтезу дедуктивних векторів, який відрізняється технологічною простотою паралельної обробки таблиць істинності та дає можливість створювати структурно-логічні умови для моделювання несправностей у цифрових системах вентильного, реєстрового та системного рівнів опису.

4.1 Вихідні дані

Вихідною інформацією до виконання векторного алгоритму є таблиця істинності та вхідний набір даних. Основні пункти алгоритму розглядаються на наступній таблиці, представленій на рис. 4.1, де показані процедури синтезу векторів для двох вхідних наборів 10011 і 10001, позначених червоним кольором.

4.2 Опис алгоритму

Алгоритм синтезу дедуктивних векторів на прикладі вхідного набору 10011:

- 1) Завдання вхідного вектора та стану виходу у рамках таблиці істинності.
- 2) Інвертування стовпців таблиці відповідно до одиничних значень змінних вхідного вектора. Результат інверсії показаний у стовпцях (6)-(9).
- 3) Визначення рядків-адрес у даних стовпцях (6)-(9), що відповідають одиничним значенням координат вектора-стовпця 5 станів виходу.
- 4) Формування нульових (одиничних) координат, якщо стан виходу функції дорівнює 1 (0), в дедуктивному векторі 10 по всіх векторах, зазначених зеленим кольором, де всі інші координати рівні одиниці (нулю).

X_1	X_2	X_3	X_4	F	\bar{X}_1	X_2	X_3	\bar{X}_4	L	X_1	X_2	X_3	X_4	F	\bar{X}_1	X_2	X_3	X_4	L	
0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1
0	0	1	0	0	1	0	1	1	1	0	1	0	0	1	0	1	0	0	0	0
0	0	1	1	0	1	0	1	0	1	0	0	1	1	0	1	0	1	1	0	0
0	1	0	0	0	1	1	0	1	1	0	0	1	0	0	1	1	0	0	0	0
0	1	0	1	1	1	1	0	0	1	1	0	1	1	1	1	0	0	1	0	0
0	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	0
0	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1
1	0	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0
1	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
1	0	1	1	0	0	0	1	0	1	0	1	0	1	0	0	0	1	1	0	0
1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1	1
1	1	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1	0	0	0	0
1	1	1	1	1	0	1	1	0	1	0	1	1	1	1	0	1	1	1	0	0

Рисунок 4.1 – Синтез дедуктивних векторів для функції від 4-х змінних

5) Визначення дедуктивної ДНФ для заданого вхідного набору (10011 та 10000) за дедуктивним вектором, що необхідно для верифікації алгоритму:

$$\begin{aligned}
 L &= [\bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4 \vee \bar{X}_1X_2\bar{X}_3X_4 \vee X_1\bar{X}_2\bar{X}_3X_4 \vee X_1X_2X_3X_4] \wedge (T=1001) = \\
 &= (X_1\bar{X}_2\bar{X}_3X_4 \vee X_1X_2\bar{X}_3\bar{X}_4 \vee \bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4 \vee \bar{X}_1X_2X_3\bar{X}_4) \oplus 1 = \\
 &= \bar{X}_1X_2\bar{X}_3 \vee \bar{X}_1X_4 \vee X_2X_4 \vee X_1X_3 \vee \bar{X}_2X_3 \vee X_3X_4 \vee X_1\bar{X}_2\bar{X}_4.
 \end{aligned}$$

Формування дедуктивного вектора для вхідного набору 10000 показано в стовпцях (11)-(20), а верифікація подана в такому вигляді:

$$\begin{aligned}
 L &= [\bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4 \vee \bar{X}_1X_2\bar{X}_3X_4 \vee X_1\bar{X}_2\bar{X}_3X_4 \vee\vee X_1X_2X_3X_4]\wedge(T=1000) = \\
 &= (X_1\bar{X}_2\bar{X}_3\bar{X}_4 \vee X_1X_2\bar{X}_3X_4 \vee \bar{X}_1\bar{X}_2\bar{X}_3X_4 \vee\vee \bar{X}_1X_2X_3X_4)\oplus 0= \\
 &= X_1\bar{X}_2\bar{X}_3\bar{X}_4 \vee X_1X_2\bar{X}_3X_4 \vee \bar{X}_1\bar{X}_2\bar{X}_3X_4 \vee\vee \bar{X}_1X_2X_3X_4].
 \end{aligned}$$

Поданий алгоритм синтезу дедуктивних векторів відрізняється технологічною простотою паралельної обробки таблиць істинності. Це дає можливість створювати структурно-логічні умови для моделювання несправностей у цифрових системах вентильного, реєстрового та системного рівнів опису.

Метрика технологічності структур даних визначається простотою алгоритму. Ідеальними є структури даних, які забезпечують один рядок коду виконання алгоритму їх обробки.

У випадку алгоритму синтезу дедуктивного вектора для заданого вхідного набору з 1(0)-станом виходу з метою транспортування несправностей через функціональний елемент виконуються лише дві процедури:

1) Інвертування стовпців таблиці істинності за одиничними координатами вхідного набору – отримання дедуктивної таблиці вхідних значень.

2) Формування 0(1)-координат дедуктивного вектора за ліченими адресами інвертованої таблиці, що відповідають 1(0)-координатам вектора вихідних станів функціональності – отримання вектора вихідних значень.

Алгоритм можна звести до однієї процедури – отримання дедуктивної таблиці істинності для вхідного набору заданої функціональності.

Синтез дедуктивної таблиці істинності за правилом:

$$L=T\oplus F \rightarrow L_i = T_i \oplus F_i.$$

Тут T – тестовий набір даних (рядок таблиці істинності) за вхідними та вихідним координатами;

T_i – координата вхідного набору;

F – таблиця істинності для заданої функціональності;

F_i – вектор стовпець таблиці істинності;

L – дедуктивна таблиця істинності;

L_i – вектор стовпець дедуктивної таблиці істинності;

\oplus – координатно-векторна операція над стовпцями.

Для пояснення останнього алгоритму в один рядок коду додано стовпець R (рис. 4.2), який позначає стан дедуктивної таблиці істинності щодо порядку проходження вхідних наборів вихідної функціональності F .

Після отримання неупорядкованої за вхідними кодами дедуктивної таблиці істинності виконується її реструктуризація до виду початкової таблиці, але з іншим (дедуктивним) вектором виходів $R=f(F)$. Результатом – дедуктивною таблицею – слід вважати стовпці (1)–(4), (10), і навіть стовпці (12)–(15), (22). За фактом весь алгоритм зводиться до синтезу стовпця дедуктивного стану виходів $L (R)$ для вихідної таблиці істинності F з урахуванням тестового набору T .

X_1	X_2	X_3	X_4	F	\bar{X}_1	X_2	X_3	\bar{X}_4	L	R	X_1	X_2	X_3	X_4	F	\bar{X}_1	X_2	X_3	X_4	L	R	
0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0	
0	0	0	1	0	1	0	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0	1
0	0	1	0	0	1	0	1	1	1	1	0	0	1	0	0	1	0	1	0	0	0	0
0	0	1	1	0	1	0	1	0	1	1	0	0	1	0	1	0	1	1	1	0	0	0
0	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0
0	1	0	1	1	1	1	0	0	0	0	0	1	1	1	1	0	1	0	1	1	0	0
0	1	1	0	0	1	1	1	1	1	1	0	0	1	0	0	1	1	1	0	0	0	0
0	1	1	1	0	1	1	1	0	1	1	0	0	1	0	1	1	1	1	1	0	1	0
1	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
1	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0
1	0	1	0	0	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	1	1	0	1	0	0	0	0	1	1	1	0	0	0
1	1	0	0	0	0	1	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	1	0	1	0	1	1
1	1	1	0	0	0	1	1	1	1	1	0	1	1	0	0	1	1	0	0	0	0	0
1	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0	1	1	1	1	1	0	0

Рисунок 4.2 – Синтез дедуктивної таблиці істинності по $L=T \oplus F$

4.3 Порівняння результатів

Для порівняння обчислювальної складності отримання дедуктивних формул нижче наводиться аналітична класична процедура аналізу елемента OR, що оперує булевими рівняннями, для яких необхідно мати складні вирішувачі [19, 26]:

$$\begin{aligned}
& L[T = (00,01,10,11), F = (X_1 \vee X_2)] = \\
& = L\{(\bar{x}_1\bar{x}_2 \vee \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_1x_2) \wedge [(X_1 \oplus T_{t1} \vee X_2 \oplus T_{t2}) \oplus T_{t3}]\} = \\
& = (\bar{x}_1\bar{x}_2)\{(X_1 \oplus 0) \vee (X_2 \oplus 0)\} \oplus 0 \vee (\bar{x}_1x_2)\{(X_1 \oplus 0) \vee (X_2 \oplus 1)\} \oplus 1 \vee \\
& \vee (x_1\bar{x}_2)\{(X_1 \oplus 1) \vee (X_2 \oplus 0)\} \oplus 1 \vee (x_1x_2)\{(X_1 \oplus 1) \vee (X_2 \oplus 1)\} \oplus 1 = \\
& = (\bar{x}_1\bar{x}_2)(X_1 \vee X_2) \vee (\bar{x}_1x_2)(\bar{X}_1 \wedge X_2) \vee (x_1\bar{x}_2)(X_1 \wedge \bar{X}_2) \vee (x_1x_2)(X_1 \wedge X_2).
\end{aligned}$$

Природно, що обробка векторних моделей є більш технологічною процедурою, де обчислювальна складність отримання дедуктивних формул транспортування дефектів функції від n змінних дорівнює $Q=n(N+F)$. Якщо врахувати, що $N+F$, як процедури інверсії векторів N та їх подальшого складання F у вектор-функцію (об'єднання або перетин) дорівнюють числу змінних $N+F=n$, то оцінка обчислювальної складності у векторних операціях дорівнює $Q=n(N+F)=n^2$.

У цьому слід зазначити, що аналітичний алгоритм синтезу дедуктивних формул має кубічну обчислювальну складність.

4.4 Висновки до розділу 4

Таким чином, алгоритмізація методу синтезу дедуктивних векторів для моделювання та/або транспортування несправностей дозволяє обробляти функціонально складні цифрові системи RTL-рівня.

Поданий алгоритм синтезу дедуктивних векторів відрізняється технологічною простотою паралельної обробки таблиць істинності. Це дає можливість створювати структурно-логічні умови для моделювання несправностей у цифрових системах вентильного, реєстрового та системного рівнів опису.

ВИСНОВКИ

Проаналізовано стан проблеми та виконано огляд моделей, методів та технологічних рішень, орієнтованих на створення ефективних засобів, які надають можливість виконувати аналіз несправностей цифрових пристроїв. Визначено основні проблеми у галузі моделювання несправностей. Зроблено висновок, що реалізація дедуктивного моделювання на основі векторної форми опису надмірної логіки транспортування несправностей надасть можливість суттєво спростити алгоритми моделювання для обробки цифрових схем великої розмірності.

Розглянуто дедуктивне моделювання несправностей цифрових пристроїв з урахуванням удосконаленої моделі хог-відношень між логічними функціями *or*, *xor*, *and*, яка згортається у нуль-простір [41].

Наукова новизна дослідження визначається:

1) удосконаленням методу синтезу дедуктивних формул для схем великої розмірності, представлених RTL-елементами на основі опису елементів будь-якої складності у вигляді вектора вихідних станів комбінаційного пристрою для транспортування вхідних списків несправностей, що має квадратичну обчислювальну складність регістрових операцій.

2) векторною модифікацією дедуктивно-векторного методу моделювання несправностей, що використовує дедуктивні формули, із застосуванням трикутника хог-відношень.

Практика застосування дедуктивного аналізу на основі $L=T \oplus F$ визначається необхідністю діагностування деструктивних компонентів у квантових, цифрових, кіберпросторових, фізичних, технічних, енергетичних, транспортних, мережевих, соціальних, юридичних, економічних процесах та явищах.

Практична значимість результатів дослідження полягає у:

- 1) визначенні переваг векторної моделі для компактного опису процесів, явищ, функцій та структур;
- 2) відмові від аналітичних виразів, що вимагають алгоритмічно складних обчислювачів-аналізаторів, із заміною їх на векторні структури даних для опису функціональної логіки;
- 3) застосуванні координатно-векторної моделі дефектів, що не прив'язана до вхідних змінних, яка може бути використана для ефективної обробки складних логічних схем при оцінці якості синтезованих тестів;
- 4) розробці алгоритму синтезу дедуктивних векторів, який відрізняється від відомих технологічною простотою паралельної обробки таблиць істинності. Це надає можливість створювати структурно-логічні умови для моделювання несправностей у цифрових проектах вентильного, реєстрового та системного рівнів опису.

Виконано синтез дедуктивних таблиць істинності за правилом $L=T \oplus F$, який відрізняється застосуванням єдиної векторно-координатної паралельної хог-операції, що дозволяє забезпечити транспортування несправностей через функціональний елемент довільної складності.

Таким чином, запропонований синтез дедуктивної таблиці істинності за правилом $L=T \oplus F$ є найефективнішим методом. Він відрізняється від відомих застосуванням єдиної векторно-координатної паралельної хог-операції, яка забезпечує транспортування несправностей через функціональний елемент довільної складності. Повна дедуктивна матриця Q функціонального елемента довільної складності від змінних n для аналізу 2^n вхідних впливів з метою отримання якості тесту має розмірність $Q = 2^n \times (2^n + n)$.

Обробка векторних моделей є більш технологічною процедурою, де обчислювальна складність отримання дедуктивних формул транспортування

дефектів функції від n змінних дорівнює $Q=n(N+F)$. Якщо врахувати, що $N+F$, як процедури інверсії векторів N та їх подальшого складання F у вектор-функцію (об'єднання або перетин) дорівнюють числу змінних $N+F=n$, то оцінка обчислювальної складності у векторних операціях дорівнює $Q=n(N+F)=n^2$.

При цьому слід зазначити, що аналітичний алгоритм синтезу дедуктивних формул має кубічну обчислювальну складність.

Перспективи дослідження. Отримані результати можуть бути використані для вирішення завдань технічної діагностики, машинного навчання, пошуку подібності-відмінності між процесами та явищами.

Результати відображено у тезах доповіді на науковій конференції [43] (додаток Б).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Michael J. Biercuk & Thomas M. Stace. Quantum Computing's Achilles. Unavoidable errors and how to fix them. IEEE Spectrum, July 2022. P. 28–33, 46.
2. R. Versluis and C. Hagen, "Quantum computers scale up: Constructing a universal quantum computer with a large number of qubits will be hard but not impossible," in IEEE Spectrum, vol. 57, no. 4, pp. 24-29, April 2020, doi: 10.1109/MSPEC.2020.9055969.
3. M. Renovell, J. M. Portal, J. Figueras and Y. Zorian, "Testing the interconnect of RAM-based FPGAs," in IEEE Design & Test of Computers, vol. 15, no. 1, pp. 45-50, Jan.-March 1998, doi: 10.1109/54.655182.
4. G. Harutunvan, V. A. Vardanian and Y. Zorian, "Minimal March tests for unlinked static faults in random access memories," 23rd IEEE VLSI Test Symposium (VTS'05), 2005, pp. 53-59, doi: 10.1109/VTS.2005.56.
5. M. Psarakis, D. Gizopoulos, A. Paschalis and Y. Zorian, "Sequential fault modeling and test pattern generation for CMOS iterative logic arrays," in IEEE Transactions on Computers, vol. 49, no. 10, pp. 1083-1099, Oct. 2000, doi: 10.1109/12.888044.
6. M. Renovell, J. M. Portal, J. Figueras and Y. Zorian, "RAM-based FPGAs: a test approach for the logic," Proceedings Design, Automation and Test in Europe, 1998, pp. 82-88, doi: 10.1109/DATE.1998.655840.
7. M. Psarakis, D. Gizopoulos, A. Paschalis and Y. Zorian, "Sequential fault modeling and test pattern generation for CMOS iterative logic arrays," in IEEE Transactions on Computers, vol. 49, no. 10, pp. 1083-1099, Oct. 2000, doi: 10.1109/12.888044.

8. U. Reinsalu, J. Raik, R. Ubar and P. Ellervee, "Fast RTL Fault Simulation Using Decision Diagrams and Bitwise Set Operations," 2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Vancouver, BC, 2011, pp. 164-170, doi: 10.1109/DFT.2011.42.

9. R. Ubar, S. Devadze, J. Raik and A. Jutman, "Fast Fault Simulation for Extended Class of Faults in Scan Path Circuits," 2010 Fifth IEEE International Symposium on Electronic Design, Test & Applications, Ho Chi Minh City, 2010, pp. 14-19, doi: 10.1109/DELTA.2010.32.

10. U. Reinsalu, J. Raik and R. Ubar, "Register-transfer level deductive fault simulation using decision diagrams," 2010 12th Biennial Baltic Electronics Conference, 2010, pp. 193-196, doi: 10.1109/BEC.2010.5631842.

11. I. Pomeranz and S. M. Reddy, "Unspecified Transition Faults: A Transition Fault Model for At-Speed Fault Simulation and Test Generation," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 1, pp. 137-146, Jan. 2008, doi: 10.1109/TCAD.2007.907000.

12. I. Pomeranz and S. M. Reddy, "Test data compression based on output dependence," 2003 Design, Automation and Test in Europe Conference and Exhibition, 2003, pp. 1186-1187, doi: 10.1109/DATE.2003.1253793.

13. I. Pomeranz and S. M. Reddy, "Unspecified Transition Faults: A Transition Fault Model for At-Speed Fault Simulation and Test Generation," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 1, pp. 137-146, Jan. 2008, doi: 10.1109/TCAD.2007.907000.

14. I. Pomeranz and S. M. Reddy, "A synthesis procedure for flexible logic functions," Proceedings Design, Automation and Test in Europe, 1998, pp. 973-974, doi: 10.1109/DATE.1998.655995.

15. I. Pomeranz, L. N. Reddy and S. M. Reddy, "COMPACTEST: A METHOD TO GENERATE COMPACT TEST SETS FOR COMBINATIONAL CIRCUITS," 1991,

Proceedings. International Test Conference, 1991, pp. 194-, doi: 10.1109/TEST.1991.519510.

16. I. Pomeranz and S. M. Reddy, "On interconnecting circuits with multiple scan chains for improved test data compression," 17th International Conference on VLSI Design. Proceedings., 2004, pp. 741-744, doi: 10.1109/ICVD.2004.1261016.

17. I. Pomeranz and Y. Zorian, "Fault isolation for nonisolated blocks," in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 12, no. 12, pp. 1385-1388, Dec. 2004, doi: 10.1109/TVLSI.2004.837994.

18. Ruifeng Guo, I. Pomeranz and S. M. Reddy, "A fault simulation-based test pattern generator for synchronous sequential circuits," Proceedings 17th IEEE VLSI Test Symposium (Cat. No.PR00146), 1999, pp. 260-267, doi: 10.1109/VTEST.1999.766674.

19. Hahanov, Vladimir. "Cyber Physical Computing for IoT-driven Services". New York: Publ. Springer. 2018. 279 p.

20. Liu, T. Yu, T. Wang S. & Cai, S. "An efficient degraded deductive fault simulator for small-delay defects". In *IEEE Access*. 2020; Vol.8: 204855–204862. DOI: <https://doi.org/10.1109/ACCESS.2020.3037292>.

21. Chumachenko, S., Iemelianov, I., Larchenko, L. and all. "Deductive qubit fault simulation". *14th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*. 2017; p. 256–259. DOI: <https://doi.org/10.1109/CADSM.2017.7916129>.

22. Reinsalu, U., Raik, J. & Ubar, R. "Register-transfer level deductive fault simulation using decision diagrams". *12th Biennial Baltic Electronics Conference*. 2010: P. 193–196. DOI: <https://doi.org/10.1109/BEC.2010.5631842>.

23. Hacimahmud, A. V., Litvinova, E., Chumachenko, S. and all. "Quantum deductive simulation for logic functions". *IEEE East-West Design & Test Symposium (EWDTS)*, 2018; p. 1–7. DOI: <https://doi.org/10.1109/EWDTS.2018.8524619>.

24. Dobai, R. & Gramatova, E "Deductive Fault Simulation for Asynchronous Sequential Circuits," *12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools*, 2009. P. 459-464, doi: <https://doi.org/10.1109/DSD.2009.129>.

25. Gharibi, W., Litvinova, E., Chumachenko, S. and all. "Qubit-driven Fault Simulation", *IEEE Latin American Test Symposium (LATS)*. 2019; P. 1–7. DOI <https://doi.org/10.1109/LATW.2019.8704583>.

26. Hahanov, V., Wajeb Garibi, Svetlana Chumachenko, Eugenia Litvinova, Abdullayev Vugar Hacimahmud & Tariq Hama Salih, "Vector-Qubit models for SoC Logic-Structure Testing and Fault Simulation," *IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, 2021. p. 24–28, DOI: <https://doi.org/10.1109/CADSM52681.2021.9385266>.

27. Abramovici, M., Breuer, M.A. & Friedman, A.D. "Digital System Testing and Testable Design". *Comp. Sc. Press*. 1998; 652 p.

28. Pomeranz, I. & Reddy, S. M. "Forward-looking fault simulation for improved static compaction," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Oct. 2001; Vol. 20 no. 10: 1262–1265. DOI: <https://doi.org/10.1109/43.952743>.

29. Pomeranz, I. & Reddy, S. M. "Unspecified Transition Faults: A Transition Fault Model for At-Speed Fault Simulation and Test Generation," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Jan. 2008; vol. 27, no. 1: 137–146. DOI: <https://doi.org/10.1109/TCAD.2007.907000>.

30. Larshin, V. P., Lishchenko, N. V., Babiychuk, O. B. & Piteľ, Ján. "Computer-Aided Design and Production Information Support". *Herald of Advanced Information Technology. Publ. Nauka i Tekhnika*. Odessa: Ukraine. 2021; Vol. 4 No. 2: 111–122. DOI: <https://doi.org/10.15276/hait.02.2021.1>

31. Drozd, O.V., Rucinski, A., Zashcholkin, K.V., Drozd, M.O. & Sulima, Y.Y. Improving FPGA components of critical systems based on natural version redundancy. *Applied Aspects of Information Technology*. Mar. 2014; 2: 168–177. DOI: <https://doi.org/10.15276/aait.02.2021.4>

32. Viktor V. Antoniuk, Myroslav O. Drozd, Oleksandr B. Drozd. Power-Oriented Checkability and Monitoring of the Current Consumption in FPGA Projects of the Critical Applications. *Applied Aspects of Information Technology*. 2019; Vol.2 No.2: 105 – 114. DOI: 10.15276/aait.02.2019.2

33. Kovalev I., S., Drozd O. V., Rucinski A., Drozd M. O., Antoniuk V. V., Sulima Y. Y. Development of Computer System Components in Critical Applications: Problems, Their Origins and Solutions. *Herald of Advanced Information Technology*. 2020; Vol.3 No.4: 252–262. DOI: 10.15276/hait.04.2020.4

34. D.V. Efanov, V.V. Sapozhnikov, and VI.V. Sapozhnikov “Synthesis of Self-Checking Combination Devices Based on Allocating Special Groups of Outputs”, *Automation and Remote Control*, 2018, Vol. 79, Issue 9, pp. 1609-1620, doi: 10.1134/S0005117918090060.

35. A. Drozd, S. Antoshchuk, “New on-line testing methods for approximate data processing in the computing circuits,” 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Prague, Czech Republic, 2011, pp. 291–294. DOI:10.1109/IDAACS.2011.6072759

36. M. Drozd, A. Drozd, “Safety-Related Instrumentation and Control Systems and a Problem of the Hidden Faults,” *The 10th International Conference on Digital Technologies 2014*, Zhilina, Slovak Republic, 2014, pp. 137–140. DOI: 10.1109/DT.2014.6868692

37. D.V. Efanov, V.V. Sapozhnikov, and VI.V. Sapozhnikov “Organization of a Fully Self-Checking Structure of a Combinational Device Based on Searching for Groups

of Symmetrically Independent Outputs”, Automatic Control and Computer Sciences, 2020, Vol. 54, Issue 4, pp. 279-290, doi: 10.3103/S0146411620040045.

38. A. Drozd, J. Drozd, S. Antoshchuk, V. Nikul, M. Al-dhabi, “Objects and Methods of On-Line Testing: Main Requirements and Perspectives of Development,” Proc. IEEE East-West Design & Test Symposium, Yerevan, Armenia, 2016, pp. 72 – 76. DOI: 10.1109/EWDTS.2016.7807750

39. O. Drozd, K. Zashcholkin, O. Martynyuk, O. Ivanova, J. Drozd. Development of Checkability in FPGA Components of Safety-Related Systems. CEUR Workshop Proceedings, vol. 2762, pp. 30-42 (2020). Online <http://ceur-ws.org/Vol-2762/paper1.pdf>

40. D.V. Efanov, V.V. Sapozhnikov, and V.I. Sapozhnikov “Boolean-Complement Based Fault-Tolerant Electronic Device Architectures”, Automation and Remote Control, 2021, Vol. 82, Issue 8, pp. 1403-1417, doi: 10.1134/S0005117921080075.

41. Hahanova A. V. Developing method of vector synthesis deductive logic for computer systems fault analysis. Herald of Advanced Information Technology. 2022; Vol. 5 No.2: 102–112. DOI: <https://doi.org/10.15276/hait.05.2022.8>

42. Gartner Top 10 Strategic Technology Trends 2023. <https://www.gartner.com/en/articles/gartner-top-10-strategic-technology-trends-for-2023>

43. Гайденко К. А., Хаханова Г. В., Чумаченко С. В. Синтез дедуктивних формул транспортування несправностей // Матеріали IV Міжнародної наукової конференції «Розвиток наукової думки постіндустріального суспільства: сучасний дискурс». 18.11.2022. Львів, Україна. С. 229-231. Імпакт-фактор Google Scholar <https://archive.mcmd.org.ua/index.php/conference-proceeding/issue/view/18.11.2022>