

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

УДК 004.94

До захисту допускається

Завідувач кафедри

_____ Скарга-Бандурова І.С.

« ____ » _____ 2019 р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Методи вдосконалення алгоритмів пошуку дефектів у цифрових системах

Освітньо-кваліфікаційний рівень "Магістр"

Спеціальність 123 – "Комп'ютерна інженерія"

Науковий керівник роботи:

(підпис)

В.С. Кардашук

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я.О. Критська

(ініціали, прізвище)

Студент:

(підпис)

І.В. Касабуцька

(ініціали, прізвище)

Група:

КІ-17дМ

Севєродонецьк 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерної інженерії

Освітньо-кваліфікаційний рівень магістр

Напрямок підготовки 123 Комп'ютерна інженерія

(шифр і назва)

Спеціальність _____

(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____

І.С. Скарга-Бандурова

« _____ » _____ 2019 р.

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Касабуцька Ірина Валеріївна

(прізвище, ім'я, по батькові)

1. Тема роботи Методи вдосконалення алгоритмів пошуку дефектів у цифрових системах

керівник проекту (роботи) Кардашук В.С., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом вищого навчального закладу від "18" 10 2018 р. № 220/48

2. Термін подання студентом роботи 11.01.2019

3. Вихідні дані до роботи Матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Існуючі методи та засоби пошуку дефектів.

2. Класифікація та побудова алгоритмів діагностування.

3. Методи вдосконалення пошуку дефектів у цифрових системах.

4. Охорона праці.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Презентація

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	Кардашук В.С., доц..		
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О., ст. викладач		

7. Дата видачі завдання _____

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання	15.12.2018	
2	Пошук та аналіз наукових статей	08.01.2018 - 15.03.2018	
3	Дослідження існуючих методів і засобів пошуку дефектів в цифрових системах	16.03.2018 - 25.06.2018	
4	Аналіз методів і засобів пошуку дефектів в цифрових системах	26.06.2018 - 01.09.2018	
5	Оформлення першого розділу	02.09.2018 - 20.10.2018	
6	Розробка рекомендацій щодо подальшого використання досліджених методів	21.10.2018 - 01.12.2018	
7	Оформлення пояснювальної записки	02.12.2018- 09.01.2019	
8	Захист дипломного проекту	18.01.2019	

Студент _____

(підпис)

Касабуцька І. В.

(прізвище та ініціали)

Керівник _____

(підпис)

Кардашук В. С.

(прізвище та ініціали)

АНОТАЦІЯ

Касабуцька І.В. Методи вдосконалення алгоритмів пошуку дефектів у цифрових системах.

Виконано дослідження методів побудови алгоритмів у цифрових системах на кристалах. Розглянуті основи технічного та тестового діагностування, структурні алгоритми пошуку дефектів, класифікація та побудова алгоритмів діагностування, що використовують таблиці несправностей. Проведений аналіз існуючих несправностей в цифрових системах.

Запропонований дедуктивний метод структурного аналізу, дедуктивно-паралельний метод моделювання графових структур, дедуктивний аналіз та синтез дедуктивних компонентів для SoC. Запропоновані методи призначені для зниження трудомісткості, скорочення часових витрат і підвищення ефективності процедури пошуку несправностей у цифрових системах, в тому числі у цифрових системах на кристалах.

Ключові слова: діагностика, алгоритм, дедуктивний метод, дедуктивно-паралельний метод, SiP, SoC, дефекти, тестування, цифрові системи, синтез, таблиця істинності.

THE ABSTRACT

Kasabutska I.V. Methods for improving troubleshooting algorithms in digital systems.

The research of methods for constructing algorithms in digital systems on crystals. The basics of technical and test diagnostics, structural troubleshooting algorithms, classification and construction of diagnostic algorithms using fault tables are considered. The analysis of existing faults in digital systems.

A deductive structural analysis method, a deductive-parallel method for modeling graph structures, a deductive analysis, and a synthesis of deductive components for SoC are proposed. The proposed methods are designed to reduce labor intensity, reduce time costs and improve the efficiency of the troubleshooting procedure in digital systems, including digital systems on crystals.

Key words: diagnostics, algorithm, deductive method, deductive-parallel method, SiP, SoC, defects, testing, digital systems, synthesis, truth table.

ПЕРЕЛІК СКОРОЧЕНЬ

ДЕ	Діагностичний експеримент
ДІ	Діагностична інформація
ДНФ	Диз'юнктивна нормальна форма
ДФ	Дедуктивна функція
ЕП	Елементарна перевірка
ЗЗ	Зворотні зв'язки
ЗР	Збіжні розгалуження
КЕД	Класи еквівалентних дефектів
КТ	Контрольна точка
ОД	Об'єкт діагностування
ПЕ	Примітивний елемент
СА	Сигнатурний аналізатор
ТЕЗ	Типовий елемент заміни
ТН	Таблиця несправностей
ТФН	Таблиця функцій несправностей
ЦП	Цифровий пристрій
SiP	System-on-Pack
SoC	System-on-Chip

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ПОШУКУ ДЕФЕКТІВ У ЦИФРОВИХ СИСТЕМАХ	11
1.1 Основи технічного діагностування	11
1.2 Аналіз несправностей цифрових систем	14
1.3 Аналіз причин несправностей у цифрових системах	16
1.4 Аналіз існуючих систем тестування цифрових систем	19
1.4.1 LogicVision	19
1.4.2 Synopsys	20
1.4.3 Mentor Graphics (Siemens)	22
1.4.4 Cadence	24
1.4.5 Узагальнення аналізу систем тестування цифрових систем	24
1.5 Аналіз стратегій тестування SiP і SoC	25
1.6 Висновки до розділу 1 та постановка задачі дослідження	26
РОЗДІЛ 2 КЛАСИФІКАЦІЯ ТА ПОБУДОВА АЛГОРИТМІВ ДІАГНОСТУВАННЯ	27
2.1 Основні визначення в алгоритмах діагностування	27
2.2 Властивості таблиць функцій несправності	28
2.3 Побудова умовного алгоритму пошуку дефектів у вигляді дерева	30
2.4 Алгоритми діагностики	33
2.5 Умовний алгоритм пошуку дефектів	34
2.6 Висновки до розділу 2	37
РОЗДІЛ 3 СТРУКТУРНІ АЛГОРИТМИ ПОШУКУ ДЕФЕКТІВ В СИСТЕМАХ ДІАГНОСТУВАННЯ	38
3.1 Основні визначення в структурних алгоритмах пошуку дефектів	38
3.2 Методи стиску двійкових послідовностей. Сигнатурний аналіз	40
3.3 Зондовий алгоритм пошуку дефектів. Зворотнє простеження	43
3.4 Умовний зондовий алгоритм пошуку дефекту	49
3.5 Висновок до розділу 3	60
РОЗДІЛ 4 ВИКОРИСТАННЯ ДЕДУКТИВНО-ПАРАЛЕЛЬНОГО МЕТОДУ ДЛЯ ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ У ЦИФРОВИХ СИСТЕМАХ	62
4.1 Узагальнена модель дедуктивно-паралельного методу аналізу несправностей	62
4.2 Дедуктивний і дедуктивно-паралельний методи структурного аналізу схеми та моделювання графових структур	75
4.3 Дедуктивний аналіз та синтез дедуктивних компонентів для SoC	80

4.4 Висновки до розділу 4.....	85
5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ.....	86
5.1 Загальні питання з охорони праці.....	86
5.2 Правові та організаційні основи охорони праці.....	87
5.3 Аналіз стану умов праці.....	87
5.3.1 Вимоги до приміщень.....	87
5.3.2 Вимоги до організації місця праці.....	88
5.3.3 Навантаження та напруженість процесу праці.....	88
5.4 Виробнича санітарія.....	89
5.4.1 Аналіз небезпечних та шкідливих факторів при роботі з персональним комп'ютером.....	90
5.4.2 Пожежна безпека.....	91
5.4.3 Електробезпека.....	92
5.5 Гігієнічні вимоги до параметрів виробничого середовища.....	92
5.5.1 Мікроклімат.....	92
5.5.2 Освітлення.....	93
5.6 Вентилювання.....	94
5.7 Охорона навколишнього природного середовища.....	95
5.8 Висновки до розділу 5.....	96
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	97
ПЕРЕЛІК ПОСИЛАНЬ.....	100
ДОДАТОК – Презентація.....	105

ВСТУП

Якими би досконалими не були сучасні цифрові системи, проблеми його діагностування залишаються актуальними. Успішність проведення технологічного процесу діагностики та налагодження залежить від правильного вибору послідовності при виконанні технологічних переходів, тобто від знання і володіння методами пошуку дефектів. При достатньому різноманітті заданих допустимих елементарних перевірок можна отримати різні за якістю алгоритми діагностування, що лежать в основі цих методів. Однак найбільший практичний сенс мають завдання побудови оптимальних алгоритмів діагностування. Дані алгоритми задовольняють екстремальному (часто мінімальному) значенню цільової функції, що кількісно характеризує ту чи іншу якість алгоритму в цілому.

Розробка методів побудови оптимальних діагностичних алгоритмів є важливим компонентом вдосконалення засобів обслуговування систем. Це пов'язано безпосередньо зі зростанням їхньої складності і підвищенням вимог до ефективності їх роботи.

Домінування цифрових систем на кристалах (System-on-Chip – SoC) та цифрових систем в пакетах (System-on-Pack – SiP) пояснюється тим, що розвиток високих технологій дозволив об'єднати цифрові, аналогові, радіочастотні та навіть компоненти мікроелектромеханічних систем (Microelectromechanical systems – MEMS) в одному пристрої – чіпі, що зменшує загальний розмір системи та підвищує її продуктивність [1]. На сьогоднішній день створення мініатюрних, спеціалізованих, економічних мобільних та швидкодіючих цифрових виробів є дуже затребуваним у всьому світі. Тестування SoC і SiP необхідне для забезпечення відсутності помилок в продукції через невідворотні недоліки виробництва. Фактично тестова діяльність відіграє ключову роль у процесі виробництва інтегральних мікросхем завдяки її здатності відрізнити хороші пристрої від несправних перед доставкою кінцевим споживачам, а також для покращення сервісного обслуговування. В основному тестування складається з двох процесів: генерації тестів і тестового застосування. Виробництво тестів означає процес надання відповідних випробувальних подразників, тоді як тестова програма відноситься до процесу застосування цих тестових подразників до входів схеми та аналізу його відповідей. Адаптація існуючих технологій тестування SoC [2], а також розробка нових технологій тестування SiP на сьогоднішній час є пріоритетною проблемою серед вчених та дослідників design-and-test community. Актуальність дослідження полягає в необхідності значного підвищення швидкодії засобів моделювання та генерації тестів для структурно- та функціонально-складних цифрових систем, що імплементуються у кристали програмованої логіки.

Питання застосування методів вдосконалення алгоритмів пошуку дефектів у цифрових системах розглядається в дослідженнях вітчизняних та зарубіжних вчених: В.І. Хаханов, І.В. Хаханова, Є.І. Літвінова, О.А. Гузь, С.В. Бахвалов, Амі Городецький, Ерік Ларссон, Гуннар Карлссон, Урбан Інгельссон та інші.

Зазначені вище автори займаються розглядом підходів до вирішення проблеми з різних точок зору, дякуючи чому їх праці гармонійно доповнюють одна одну. На жаль, вузькоспрямованість проблематики обумовлює те, що даною проблемою займається невелика кількість вчених, і всі їх роботи так чи інакше базуються одна на одній (наприклад, роботи В.І. Хаханова, І.В. Хаханової, Є.І. Літвінової та О.А. Гузь).

З кожним роком вдосконалюються програмуємі логічні інтегральні схеми (далі – ПЛІС) та цифрові системи на кристалах, тому існуючі рішення вдосконалення алгоритмів пошуку дефектів у цифрових системах, що вже були запропоновані вченими, втрачають свою актуальність. Тому для того, щоб технології Hardware/Software Design and Test не застрягли на одному місці, необхідно постійно займатися пошуком нових рішень.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконувалася протягом 2017-2018 рр. згідно з планами науково-дослідних робіт кафедри комп'ютерних наук та інженерії в межах НДР «Дослідження у галузі вищої освіти і практики інституційної співпраці» (№ ДР 0113U002236).

Мета і завдання дослідження. В даному дипломному проєкті проведено дослідження методів вдосконалення алгоритмів пошуку дефектів у цифрових системах. Головною задачею дослідження є аналіз існуючих засобів та методів пошуку дефектів у цифрових системах на кристалах, пошук їх головних недоліків та розробка нового алгоритму і/або методу тестування, що буде досконаліший попередників таких як послідовний метод діагностики несправностей, умовний алгоритм пошуку дефектів за допомогою побудови дерева, автоматичні системи тестування фірм Logic Vision, Synopsys, Mentor Graphics, Cadence.

Об'єкт дослідження – процеси пошуку дефектів у цифрових системах.

Предмет дослідження – методи побудови алгоритмів пошуку дефектів у цифрових системах для зниження трудомісткості, скорочення часових витрат і підвищення ефективності процедури пошуку дефектів у цифрових системах на кристалах.

Методи дослідження. Для вирішення поставлених завдань було використано метод дедуктивно-паралельного виявлення несправностей. Був змодульований симулятор несправних примітивів, що орієнтований на створення вбудованих засобів дедуктивно-паралельного моделювання, що в порівнянні з програмною реалізацією підвищує швидкодію аналізу в 10-1000 разів. Також було використано метод паралельного дедуктивного моделювання графових структур цифрових систем, що завдяки виключенню операцій над елементами множини та

збільшенню пам'яті для зберігання матриці ідентифікаторів вершин дозволяє підвищити швидкодію аналізу вершин графу, що мають багато попередників, у десятки та навіть сотні разів.

Наукова новизна магістерської роботи полягає в дослідженні методів побудови алгоритмів пошуку дефектів у цифрових системах на кристалах. На основі дослідження вироблені рекомендації щодо подальшого використання досліджених методів.

Публікації. По даній роботі опубліковано дві наукові праці:

1. VIII Всеукраїнська науково-практична конференція "Електронні апарати та системи. Проблеми створення. Перспективи розвитку", м. Сєверодонецьк. Назва статті: "Методи вдосконалення тестування цифрових систем на кристалах" [3].

2. Вісник Східноукраїнського національного університету ім. В. Даля. 2018. №.6 (247). С. 61-68. (фахове видання). Науковий журнал, м. Сєверодонецьк. Назва статті: "Методи вдосконалення тестування цифрових систем" [4].

Структура і обсяг роботи. Магістерська робота складається зі вступу, п'ятьох розділів, висновку, списку використаних джерел і додатку. Робота викладена на 114 сторінках машинописного тексту, містить 88 сторінок основного тексту, 52 рисунка, 19 таблиць, 1 додаток на 10 сторінках. Бібліографічний список включає 66 найменувань.

РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ПОШУКУ ДЕФЕКТІВ У ЦИФРОВИХ СИТЕМАХ

1.1 Основи технічного діагностування

Технічна діагностика – це область знань, що займається вивченням технічного стану об'єктів в даний момент часу [5].

Надійність – це здатність технічного об'єкта виконувати задані функції, зберігаючи в часі значення встановлених експлуатаційних показників у заданих межах, що відповідають заданим режимам та умовам експлуатації, ремонту, технічного обслуговування, транспортування, зберігання.

Теорія надійності є основою прогнозування технічного стану об'єкта, тобто визначення технічного стану об'єкта з наперед заданою вірогідністю на майбутній період часу.

На рисунку 1.1 зображений зв'язок технічної діагностики з іншими дисциплінами, що забезпечують її.

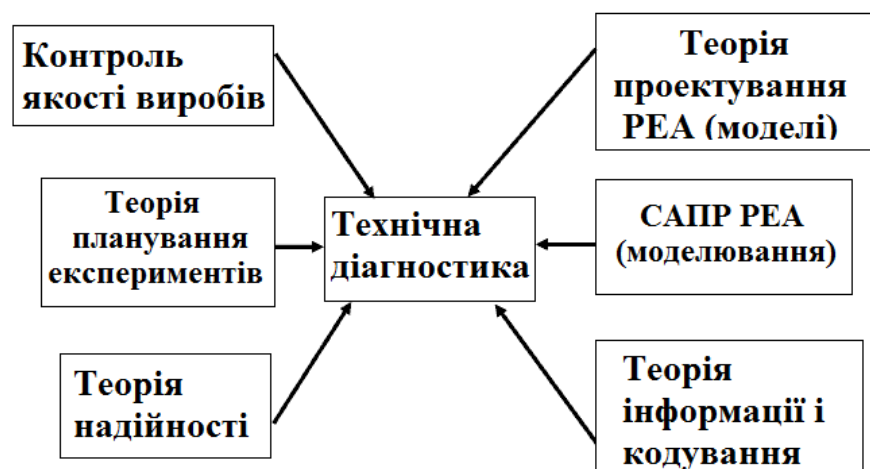


Рисунок 1.1 – Технічна діагностика, зв'язок з дисциплінами

Об'єктом технічного діагностування є виріб або його складова, технічний стан якої треба виявити. Для радіоелектронної апаратури це пристрої обчислювальної техніки (комп'ютери, мережеві обладнання, периферійні пристрої), мікросхеми, типові елементи заміни, комбінаційні схеми та комп'ютерні мережі.

Види технічного стану поділяються на [5]:

— справність – технічний стан, за якого об'єкт відповідає всім вимогам нормативно-технічної документації;

— працездатність – технічний стан, за якого об'єкт спроможний виконувати задані функції, зберігаючи значення основних технічних параметрів в межах, що задані нормативно-технічною документацією;

— правильне функціонування – спроможність об'єкта виконувати задані алгоритми функціонування при використанні його за призначенням.

Поняття справності та працездатності характеризують різні аспекти технічного стану об'єкта, а не є підмножинами один одного. Справний об'єкт може дати відмову, тобто втратити працездатність. Наприклад, справний технічно об'єкт, наприклад, комп'ютер може перестати працювати ("зависнути" абощо) через неправильні налаштування програмного забезпечення або неправильних дій користувача. Несправний об'єкт може бути працездатним, наприклад, якщо не горить лампочка контролю харчування електричної мережі. Стан правильного функціонування є підмножиною працездатного стану.

На рисунку 1.2 зображено за допомогою діаграм Ейлера-Вена, до якої множини належить той чи інший вид технічного стану, де СС – справний стан, П – працездатність, ПФ – правильне функціонування.

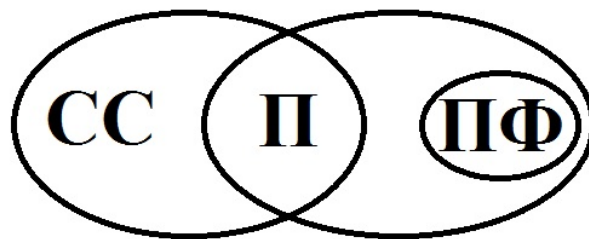


Рисунок 1.2 – Діаграма Ейлера-Вена "Види технічного стану"

Дефект є фізичною причиною зміни технічного стану об'єкта. Дефектом є фізичне пошкодження об'єкта, яке призводить до втрати працездатності (відмови). Математичною моделлю дефекту стосовно радіоелектронних об'єктів є несправність.

Щоб виявити стан об'єкту необхідно провести технічне діагностування. Результатом є висновок про технічний стан об'єкту з зазначенням, за необхідності, виду та місця знаходження дефекту.

Технічне діагностування поділяється на:

— функціональне (здійснюється під час функціонування об'єкту, та на яких подаються тільки робочі впливи) [6];

— тестове (на об'єкт подаються спеціальні впливи виключно з метою діагностування) [7].

Контроль технічного стану – це технічне діагностування з метою визначення тільки виду технічного стану об'єкта.

Тест для контролю технічного стану називається перевіряючий тест. Перевіряючий тест розрізняє справний (працездатний) стан від безлічі несправних (непрацездатних) станів.

Технічне діагностування з метою визначення місця і, при необхідності, причини та види дефекту, характеризується як пошук дефекту.

Тест, призначений для пошуку дефекту, називається тестом пошуку дефекту. Тест пошуку дефекту дозволяє розрізнити безліч несправних (непрацездатних) станів між собою.

Системою технічного діагностування називається сукупність засобів і об'єкту діагностування та, за необхідністю, виконавців, що підготовлені до діагностування або здійснюють його відповідно до алгоритмів діагностування.

Засобами технічного діагностування є сукупність апаратних та програмних засобів, за допомогою яких здійснюється технічне діагностування.

Засоби технічного діагностування поділяються на [8]:

— зовнішні – засоби діагностування, що знаходяться окремо від конструкцій об'єкта діагностування;

— вбудовані – засоби діагностування, що знаходяться в одній конструкції з об'єктом діагностування.

Для того, щоб коректно провести технічне діагностування цифрової системи треба мати алгоритм. Алгоритмом технічного діагностування називають сукупність приписів про порядок проведення діагностування.

Система тестового діагностування є найпоширенішою з систем технічного діагностування, під час якої на об'єкт подаються тестові впливи.

Системи тестового діагностування поділяються на:

— автоматизовані – діагностування проводиться за допомогою автоматизованих засобів та частково за участі оператора;

— автоматичні – діагностування проводиться без участі оператора.

На рисунку 1.3 зображений принцип технічного діагностування об'єкта за допомогою зовнішніх засобів тестового діагностування.

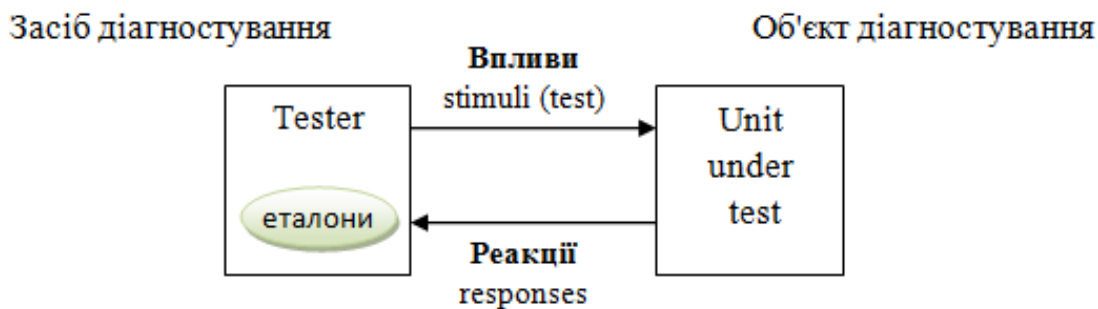


Рисунок 1.3 – Принцип тестового діагностування

При діагностуванні реального пристрою (реального продукту) пристрій вважається справним (працездатним) доти, доки в ньому не знайдені несправності.

Принципово є неможливим здійснити перевірку реального пристрою в усіх режимах на всіх типах даних.

Реальний пристрій може бути скорегований необмежену кількість разів.

Спосіб тестування реального пристрою тісно пов'язаний з процесом його розробки.

1.2 Аналіз несправностей цифрових систем

Залежно від того, як помилки виявляються в часі, можна виділити два типи помилок: постійні та непостійні помилки.

Термін «постійний збій» означає наявність несправності, що постійно впливає на функціональну поведінку системи (мікросхеми, масиву або плати).

Прикладами постійних (також званих твердими або жорсткими несправностями) є:

- неправильні з'єднання між інтегральними схемами (ІС), платою, треками і т.д. (наприклад, відсутні з'єднання або короткі замикання).
- зламані компоненти або частини компонентів.
- неправильні маски ІС, внутрішні з'єднання кремній-метал або метал-упаковка (що є виробничою проблемою).
- помилки функціонального проектування (реалізація логічної функції некоректна).

Оскільки постійні помилки постійно впливають на логічні значення в системі, їх легше виявити, ніж непостійні помилки, що описані нижче.

Непостійні помилки (несправності) присутні тільки певну частину часу. Вони відбуваються в випадкові моменти і впливають на функціональну поведінку системи протягом

кінцевих, але невідомих періодів часу. Внаслідок цієї випадкової появи важко виявити і локалізувати непостійні несправності. Якщо така несправність не впливає на систему, то система працює нормально.

Непостійні несправності можна розділити на дві групи за різним походженням: короткочасні і переривчасті несправності.

Короткочасні збої викликані умовами навколишнього середовища, такими як космічні промені, α -частинки, забруднення, вологість, температура, тиск, вібрація, коливання електроживлення, електромагнітні перешкоди, статичні електричні розряди і контури заземлення.

Короткочасні збої важко виявити через їх неясний вплив на логічні значення в системі. Помилки в оперативній пам'яті (RAM - ОЗП), що виникають при короткочасних збоях, часто називають м'якими помилками. Вони вважаються неповторюваними, і передбачається, що не було завдано постійного збитку осередку пам'яті. Випромінювання з α -частинками вважається основною причиною м'яких помилок [9].

Переривчасті несправності викликані умовами, не пов'язаними з навколишнім середовищем, такими як ненадійні з'єднання, знос або старіння компонентів (загальне припущення полягає в тому, що при переході від нормального функціонування до зношення можуть виникати переривчасті несправності), критичним часом (небезпеки і умови гонки, наприклад, може бути викликано помилками конструкції), змінами опору і ємності (значення резистора і конденсатора можуть відхилитися від свого заданого значення з самого початку або з часом, що може привести до збоїв синхронізації), фізичними порушеннями і шумом (шум заважає сигналам в системі).

Характеристика переривчастих несправностей полягає в тому, що вони поведуться як постійні несправності під час відмови, викликаного переривчастою несправністю. На жаль, час, протягом якого переривчаста несправність впливає на систему, зазвичай дуже малий в порівнянні з часом застосування тесту, розробленого для постійних несправностей, що зазвичай становить кілька секунд. Цю проблему можна вирішити, постійно повторюючи тест або змушуючи непостійну несправність стати постійною. Перехід непостійних несправностей до постійних несправностей може зайняти години, дні або місяці, і тому повинен бути прискорений. Це може бути досягнуто шляхом забезпечення певних умов стресу навколишнього середовища (температура, тиск, вологість і т.д.). Одна з проблем, пов'язаних із застосуванням умов стресу, полягає в тому, що можуть виникати нові несправності, що викликають додаткові несправності.

В контексті розгляду цифрових виробів важливо диференціювати типи дефектів за ієрархією: одиночні (single faults) [10] або кратні несправності (multiple faults) – для логічних модулів та пам'яті [11], мікродефекти – для компонентів, що входять до складу SoC або

SiP [12]. Щодо систем на кристалах, то реалізація вбудованих тестових методів діагностування передбачає наявність інфраструктури сервісного обслуговування.

1.3 Аналіз причин несправностей у цифрових системах

Фізичні та електричні причини виникнення несправностей називаються механізмами відмови. Повний огляд механізмів відмови для напівпровідникових пристроїв, до яких також належать інтегральні схеми, поділяється на три класи (рисунк 1.4) [13]:

1. Збої електричної напруги в колі: ці збої відбуваються через неправильну конструкцію, що призводить до електричного перенапруження, або через недбале поводження, що викликає статичну пошкодження.

2. Внутрішні механізми відмови: вони притаманні самій напівпровідниковій матриці; вони включають дефекти кристалів, дислокації і дефекти обробки. Зазвичай вони виникають під час виготовлення пластин і пов'язані з дефектами в оксиді або епітаксимальному шарі.

3. Зовнішні механізми відмови: вони беруть свій початок з процесу упаковки і приєднання; їх можна віднести до етапів осадження металу, зв'язування і інкапсуляції.

Згодом процес виготовлення матриці дозріває, тим самим знижуючи частоту власних відмов, щоб стати більш домінуючою. Однак дуже складно дати точний порядок механізмів відмови; деякі домінують в певних умовах експлуатації і навколишнього середовища, інші завжди присутні, але впливають менше.

Важливим параметром механізму руйнування є E_a – енергія активації, що описує температурну залежність механізму руйнування. E_a зазвичай коливається між 0,3 і 1,5 eV. Встановлено, що температура від 125 °C до 250 °C ефективна для прожигу та не викликає незворотного пошкодження [14].

Точний вплив температури на частоту відмов (тобто точне значення E_a) дуже важко визначити, і він варіюється між виробами, пакетами і т.д. У таблиці 1.1 наведені певні експериментальні значення енергій активації найбільш важливих механізмів відмови, що описані далі.

Корозія – це електромеханічний механізм руйнування, який виникає за умови наявності потенціалів вологи і постійного струму. Іони Cl^- і Na^+ діють як каталізатор. Методи упаковки (добре ущільнення) та умови навколишнього середовища в значній мірі визначають процес корозії. КМОН-пристрої (комплементарна структура метал-оксид-напівпровідник) більш сприйнятливі через їхнє низьке розсіювання потужності.

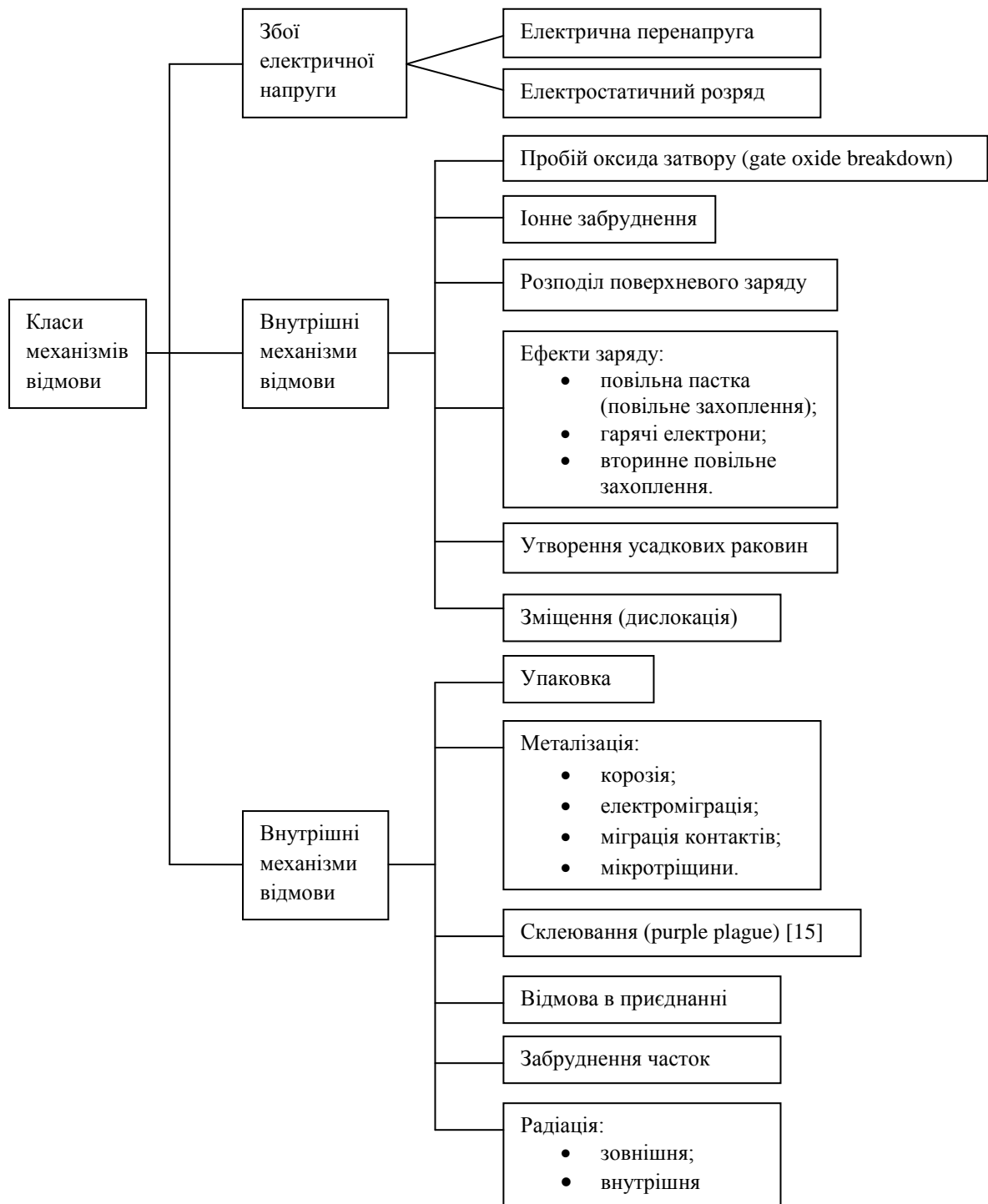


Рисунок 1.4 – Класифікація механізмів відмови

Електроміграції відбуваються в треках (лініях) металізації Al (алюмінію) чіпа. Струм електронів, що протікає через Al-треки, змушує електрони стикатися з Al-зернами. Через ці зіткнення зерна зміщуються і рухаються в напрямку струму електрона. Вузька ширина ліній, висока щільність струму і висока температура є основними причинами електроміграції, що призводить до появи відкритих ліній в місцях, де щільність струму найвища.

Таблиця 1.1 – Експериментальні значення енергій активації

Механізми відмов	E_a
Корозія металізації	0,3-0,6 eV
Електролітична корозія	0,8-1,0 eV
Електроміграція	0,4-0,8 eV
Склеювання (purple plague)	1,0-2,2 eV
Іонне забруднення	0,5-1,0 eV
Легування (міграція контактів)	1,7-1,8 eV

Склеювання є механізмом руйнування, який складається в руйнуванні контактів між Au (золотими) проводами і алюмінієвими контактними площадками мікросхеми. Це викликано взаємної дифузії Au-Al, що викликає відкрите з'єднання.

Іонне забруднення викликано рухливими іонами в напівпровідниковому матеріалі і є основним механізмом руйнування для МОН-пристроїв. Іони Na^+ є найбільш мобільними через їхній малий радіус. Вони зазвичай доступні в атмосфері, поту і диханні. Іони притягуються до оксиду затвора транзистора FET, викликаючи зміну граничної напруги пристрою.

Легування також є формою міграції алюмінію з Al в Si (кремній) або з Si в Al. Залежно від глибини з'єднання і розміру контакту несправність проявляється у вигляді замкнутого з'єднання або розімкнутого контакту. Оскільки геометрія пристрою стає менше, легування стає важливішим через меншу глибину дифузії.

Радіаційне випромінювання [9] є ще одним механізмом відмови, що особливо важливий для динамічних запам'ятовуючих пристроїв з довільним доступом (DRAM). Слідові домішки радіоактивних елементів, присутніх в пакувальному матеріалі чіпа, виділяють α -частинки з енергією до 8 MeV. Взаємодія цих α -частинок з напівпровідниковим матеріалом призводить до утворення електронно-доручених пар. Генеруються електрони, рухаються через пристрій і здатні знищувати заряд, що зберігається в осередку DRAM, що призводить до втрати його інформації. Це основна причина м'яких помилок в DRAM. Поточні дослідження показали, що статичні запам'ятовувальні пристрої з довільним доступом (SRAM) високої щільності також страждають від м'яких помилок, викликаних α -частинками [16, 17].

1.4 Аналіз існуючих систем тестування цифрових систем

Недавній прогрес в технології напівпровідникових процесів компанії, що розробляють і виробляють передові продукти, швидко переходять до дуже глибокої субмікронною (very deep submicron – VDSM) технології інтегральних схем (IC). Цей перехід відбувається тому, що технологія VDSM забезпечує підвищену функціональність, більш високу продуктивність чіпа і зниження витрат [18].

Однією з перешкод для досягнення всіх переваг технології VDSM є нездатність сучасних методик проектування і випробувань йти в ногу з постійним прогресом в розробці напівпровідникових процесів [18].

Компанії, які розробляють надвеликі інтегральні схеми (NBIC), дотримуються певних маршрутів проектування і входять до них маршрутів тестування. Вважається, що 60-80% зусиль команд-розробників апаратних засобів витрачається на верифікацію і налагодження. Важливим етапом є налагодження моделей рівня реєстрових передач (RTL). При цьому не існує універсальної методики для вирішення цього завдання [19].

Особливу увагу приділено налагодженню проектів на ПЛІС. Розбираються такі підходи: використання вбудованого логічного аналізатора, зовнішнє контрольно-вимірювальне обладнання та їх комбінування. Розглянуті такі існуючі системи тестування як:

- LogicVision;
- Synopsys;
- Mentor Graphics (Siemens);
- Cadence.

1.4.1 LogicVision

LogicVision, Inc. була компанією, що займається автоматизацією електронного проектування (electronic design automation – EDA [20]), що пропонує рішення для проектування на рівні чіпів, плат і систем (Design for Test – DFT [21]), а також підтримку для постачальників ASIC з Сан-Хосе, штат Каліфорнія, США. Заснована в 1992 році, вона була придбана компанією Mentor Graphics в 2009 році. Їх вбудований тест, BIST [22] і інструменти навчання прибутковості були застосовні до цифрових, аналогових і змішаних сигналів.

Компанія була заснована як LV Software в 1992 році Віноду Агарвалом, колишнім професором Університету Макгілла в області електротехніки [23]. Пізніше в тому ж році LV Software була перейменована в LogicVision. Компанія стала публічно торгувати в жовтні 2001

року на NASDAQ як LGVN, а в листопаді 2004 року вони придбали SiVerion. Засновник Винод Агарвал подав у відставку з поста голови ради директорів в листопаді 2005 року. 18 серпня 2009 року компанія Mentor Graphics з Орегона придбала LogicVision в рамках акції на загальну суму 13 мільйонів доларів США [24].

На даний момент LogicVision є одним з бізнес-підрозділів Mentor Graphics [24] і продовжує займатися розробкою та випуском нової продукції. Наприклад на Google Patents є інформація про 70 зареєстрованих патентів виробів даного підрозділу.

LogicVision розробляє програмно-апаратні системи та пов'язані з нею методології, що забезпечують незалежне від автоматизованого випробувального обладнання (Automated Test Equipments (ATE) [25]) тестування, що дозволяє проводити тестування і запуск, а також розширену діагностику відмов інтегральних мікросхем для налагодження кремнію, визначення характеристик процесу, тестування виробництва (обсягу) і діагностики системи. в інтегральній схемі; засіб для плавної передачі інформації між інтегральною схемою і її зовнішнім середовищем; і зовнішнє середовище, яка забезпечує плавну передачу користувачеві для виконання відповідного тесту і діагностики.

1.4.2 Synopsys

На офіційному сайті Synopsys [26] можна знайти всю інформацію про компанію та продукцію для силіконових кристалів, що вона виробляє, та її верифікацію.

Synopsys – американська компанія, що працює в області САПР для проектування електроніки [20]. Першим і найвідомішим продуктом Synopsys є Design Compiler, інструмент для логічного синтезу. Synopsys пропонує широкий спектр інших продуктів, що використовуються при розробці спеціалізованої інтегральної схеми. Продукти включають в себе логічний синтез, поведінковий синтез, місце і маршрут, статичний аналіз синхронізації, формальну верифікацію, імітатори мови опису апаратних засобів (SystemC, SystemVerilog/Verilog, VHDL), а також моделювання на рівні транзисторів. Симулятори включають середовища розробки та налагодження, які допомагають в розробці логіки для мікросхем і комп'ютерних систем. В останні роки Synopsys також вийшла на ринок безпеки додатків.

У книзі по ASIC-синтезу чіпів [27] розглянуті практичні методи розробки та верифікації тестування цифрових систем на кристалах, що використовує компанія Synopsys. Оптимізація проектів компанії виконується в два етапи (рисунок 1.5). Оптимізація логіки виконується спочатку шляхом структурування та вирівнювання дизайну. Результируюча структура потім відображається на ворота, використовуючи методи оптимізації відображення.

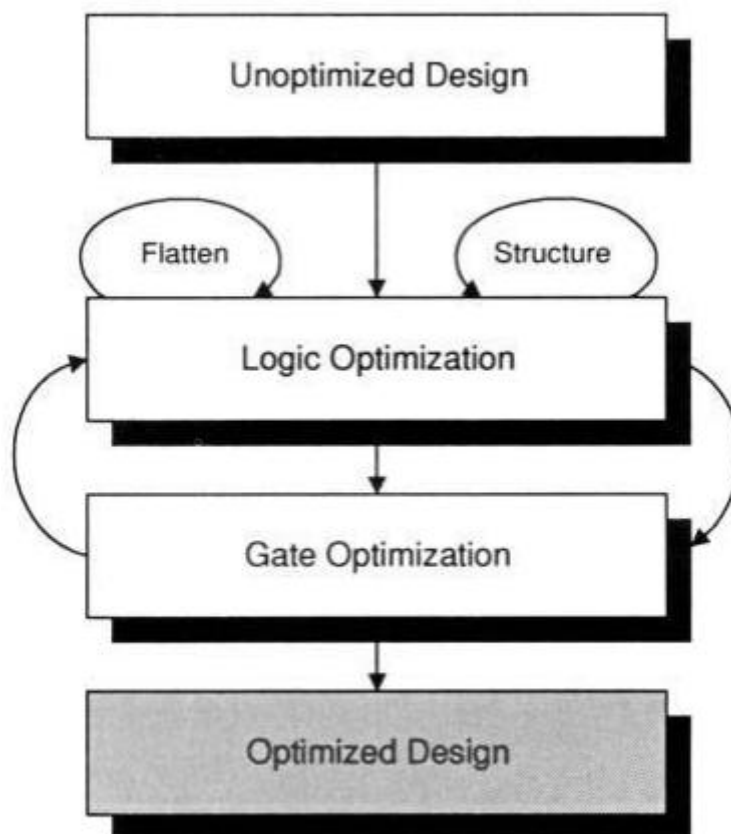


Рисунок 1.5 – Кроки оптимізації проектів Synopsys

15 березня 2004 року Synopsys анонсувала Design Compiler FPGA (DC FPGA) – новий засіб синтезу для розробників ASIC (application-specific integrated circuit – "інтегральна схема спеціального призначення"), що використовують FPGA (Field Programmable Gate Array) для протипування. Даний продукт не був надто успішним, якщо оцінювати за долею на ринку. однак їх все ще використовують для маленьких FPGA. За даними маркетологів Synopsys, 42% їх ASIC-замовників використовують FPGA для протипування [28].

За даними Synopsys DC FPGA дає наступні основні переваги:

1. Повна сумісність із засобом синтезу для ASIC – Design Compiler, тому проектувати необхідно лише один раз. DC FPGA приймає той же RTL-код, обмеження, скріпти та бібліотеки, що і Design Compiler.

2. Design Compiler включає технології оптимізації, в тому числі витяг та оптимізацію кінцевих автоматів, розподіл ресурсів, реплікацію логіки та регістрів, автогрупування критичних шляхів и т.д. Окрім того, Synopsys впровадила адаптивну технологію оптимізації. Все це забезпечує підвищення продуктивності проектів, що синтезуються, на 15% в порівнянні з засобами синтезу FPGA.

1.4.3 Mentor Graphics (Siemens)

Mentor, Siemens Business [29] – американська багатонаціональна корпорація з електротехніки та електроніки, що займається автоматизацією електронного проектування [20].

Компанія була заснована в 1981 році і продана Siemens в 2017 році.

Mentor дистрибує наступні компоненти, що пов'язані з тестуванням цифрових систем на кристалах:

- а) Автоматизація електронного проектування для:
 - 1) Повнофункціональна компоновка інтегральних схем і інструменти SDL, такі як IC Station;
 - 2) IC місце і інструмент маршруту: Olympus-SoC;
- б) Інструменти моделювання для проектування аналогових змішаних сигналів:
 - 1) ModelSim – це апаратне середовище моделювання та налагодження, в першу чергу призначена для невеликих ASIC і FPGA;
 - 2) QuestaSim – це симулятор з додатковими можливостями налагодження, призначений для складних FPGA і SoC. QuestaSim може використовуватися користувачами, що мають досвід роботи з ModelSim, оскільки він поділяє більшість загальних функцій і можливостей налагодження. Одним з основних відмінностей між QuestaSim і Modelsim (крім продуктивності/ємності) є те, що QuestaSim є механізмом моделювання для платформи Questa, яка включає в себе інтеграцію управління верифікацією, формальні технології, IP-адреси перевірки автентичності Questa, моделювання з низьким енергоспоживанням і прискорене закриття покриття. QuestaSim спочатку підтримує SystemVerilog для Testbench, UPF, UCIS, OVM / UVM, а ModelSim – немає.

Design-for-Test та напівпровідникова аналітика даних (Data Analytics) Напівпровідникова аналітика даних забезпечує високе охоплення тестами, прискорення лінійного показника текучості і поліпшення якості та надійності.

Комплексне рішення Mentor для тестування інтегральних схем, що включає кращі в своєму класі інструменти проектування для тестування і аналітики тестових даних, що допомагають забезпечити максимальне охоплення тестуванням, прискорити зміну продуктивності і підвищити якість і надійність виготовлених деталей.

Mentor використовує такі види тестування, як:

- логічний тест;
- тест пам'яті;
- тест змішаного сигналу;

— силіконове навчання.

Логічний тест [30]:

Передові методи проектування використовуються при створенні логічних частин SoC, що створює значні проблеми для досягнення високоякісного тесту на кремнію. Для вирішення цих завдань Mentor Graphics пропонує найпотужніший в галузі набір рішень для логічного тестування.

Ці рішення мають більш ніж десятиліття успішного і якісного тестування з використанням як компресії, так і методів без векторів. Разом вони забезпечують максимальну гнучкість для досягнення найбільш ефективного часу тестування в порівнянні з оптимізацією якості.

Логічні тестові рішення Tessent [31] також надають унікальну підтримку для тестування зі зменшеним числом висновків, велику підтримку для тестування конструкцій з низьким енергоспоживанням, а також передові моделі несправностей, такі як Cell-Aware.

Тест пам'яті [32]:

Рішення для тестування пам'яті Tessent надають самі передові можливості в галузі самоперевірки і відновлення пам'яті.

Ключові функції включають в себе всебічні можливості тестування і діагностики для задоволення вимог до якості нових вузлів процесів і конструкцій пам'яті, а також всебічний аналіз виправлень і можливості самовідновлення. Рішення Tessent для тестування пам'яті також забезпечують передову автоматизацію проектування, щоб гарантувати, що всі необхідні вбудовані можливості тестування і ремонту можуть бути швидко і ефективно інтегровані в проект.

Тест змішаного сигналу [33]:

Тестові рішення Tessent для змішаних сигналів не залежать від виробника і АТЕ, що відповідає зростаючій кількості інтерфейсів SerDes (Серіалізатор/Десеріалізатор [34]) і PLL (Permutation of the Last Layer) в сучасних конструкціях SoC. Характеристики з настільним обладнанням під керуванням GPIB (General Purpose Interface Bus). GPIB знижує вимоги до обладнання тестера, а PLL-стабілізатор з мікропровідним управлінням скорочує час виходу на ринок. Tessent SerdesTest і Tessent PLLTest зводять до мінімуму вимоги до обладнання тестера і знижують витрати на тестування.

Рішення Tessent для тестування пам'яті підтримують інтегральні схеми будь-якого розміру і складності, що скорочує витрати на розробку інтегральних схем і скорочує час виходу на ринок.

Силіконове навчання [35]:

Кремнієві продукти навчання Tessent підвищують продуктивність під час критичної перевірки кремнію і фаз лінійної зміни продуктивності. Продукти надають рішення для

проведення випробувань, визначення характеристик кремнію, аналізу прибутковості на основі діагностики та аналізу відмов. Спеціалізований статистичний аналіз усуває діагностичний шум і прискорює час, щоб усунути причину втрати доходів.

1.4.4 Cadence

Cadence Design Systems, Inc [36] – компанія, що займається розробкою програмного забезпечення для автоматизації проектування електронних пристроїв [20] і наданням інженерних послуг. Заснована в 1988 році в результаті об'єднання компаній SDA Systems і ECAD. Протягом багатьох років була найбільшою компанією в EDA індустрії.

Основні продукти Cadence націлені на різні типи проектування і верифікації:

– Virtuoso Platform – інструменти для проектування інтегральних схем [37]; в нього входить структурне уявлення, поведінкове моделювання (Verilog-AMS), симуляція схеми, повний макет, фізична верифікація. Використовується, в основному, для проектування аналогових схем, але так само використовується для проектування пам'яті і FPGA.

– Encounter Platform – інструменти для проектування цифрових інтегральних схем. Сюди входить проектування, тестування і синтез. [38].

– Incisive Platform – інструменти для симуляції, функціональної верифікації RTL, включаючи моделі, засновані на Verilog, VHDL і SystemC. Сюди входить формальна верифікація, перевірка формальної еквівалентності і емуляція [39].

– Allegro Platform – інструменти для спільного проектування інтегральних схем і друкованих плат.

– OrCAD/PSpice – Інструменти для невеликих компаній проектування і індивідуальних розробників.

На Google Patents є інформація про 13 зареєстрованих патентів виробів Cadence Design Systems, Inc.

1.4.5 Узагальнення аналізу систем тестування цифрових систем

Для збільшення швидкості моделювання на кілька порядків застосовуються апаратні прискорювачі, або платформи емуляції. Дані про прискорювачі, що реалізовані на спеціальних процесорах, на замовних або на комерційних ПЛІС відображені у таблиці 1.2. Всі три архітектури є масштабованими, вони дозволяють розміщувати проекти будь-якого розміру, від IP-блоків до повних систем. Важливим є тестування взаємодії на системному рівні,

підтримуване апаратними прискорювачами. Частота залежить від багатьох чинників і становить від одиниць MHz до 100 MHz (для HAPS-80) [19].

Таблиця 1.2 – Апаратні прискорювачі та платформи для емуляції

Платформа	Компанія	Мікросхеми	Кількість вентилів в проекті
Palladium	Cadence	Спеціалізовані процесори	До 2 мільярдів
Protium	Cadence	ПЛІС Xilinx Virtex-7	До 100 мільйонів
Veloce II	Mentor	Спеціалізовані ПЛІС Crystal2	От 256 мільйонів до 2 мільярдів
ZeBu Server-3	Synopsys	ПЛІС Xilinx Virtex-7	300 мільйонів, масштабується до 3 мільярдів
HAPS-80	Synopsys	ПЛІС Xilinx Virtex UltraScale	До 1,6 мільярдів

Незважаючи на великі можливості апаратних прискорювачів фірм LogicVision, Synopsys, Mentor Graphics, Cadence, системи тестування, їх засоби моделювання несправностей та синтезу тестів неприйнятні за часом обробки цифрових систем на кристалах, в яких налічується декілька мільйонів вентилів.

1.5 Аналіз стратегій тестування SiP і SoC

Стратегія тестування SiP має деякі відмінності від стратегії тестування SoC, оскільки система в пакеті інтегрує в собі компоненти від різних виробників. Використання стандарту IEEE 1500 є однією з найперспективніших стратегій для вбудованого тестування функціональних компонентів [40]. Для вбудованих інструментів тестування, з практичної точки зору, використовується стандарт IEEE 1149.1 (JTAG) [41], що успішно модифікується в такі стандарти, як IEEE 1149.7 [42] та IEEE P1687 [43] (стандарт був затверджений як новий стандарт IEEE-SA Standards Board 3 листопада 2014 року [44]).

Існуючі рішення задачі тестування та ремонту логічних компонентів цифрових систем поділяються на три класифікації:

1. Дублювання логічних елементів або областей кристалу, що приводить до подвоєння апаратної реалізації функціональності. При фіксації несправного елементу або області здійснюється перемикання за допомогою мультиплексора на справний компонент [45]. Запропоновані Xilinx - американським розробником та виробником інтегральних мікросхем програмованої логіки – моделі Field Programmable Gate Array (FPGA) також застосовуються під

час ремонту компонентів FPGA від компанії Altera. При ремонті основною одиницею вимірування є стовпчик або рядок.

2. Використання генетичних алгоритмів для діагностування та відтворення працездатності на основі автономної реконфігурації кристалу FPGA без використання зовнішніх пристроїв керування [46]. Надійність діагностування дефектів дорівнює 99%, час ремонту – 36 мілісекунд замість 660 секунд, необхідних для стандартного конфігурування проекту.

3. Відновлення працездатності кристалів FPGA не критичних за часом за допомогою заміни локальних Complex Logic Block (CLB) на надлишкові запасні компоненти запропоновано в [47,48]. Доступний рівень об'єднання CLB, що підлягає заміні, для критично важливих додатків складає приблизно тисячі логічних елементів.

1.6 Висновки до розділу 1 та постановка задачі дослідження

В першому розділі магістерської роботи здійснено огляд методів і засобів пошуку дефектів у цифрових системах. Визначені такі поняття, як технічна діагностика, надійність, справність, працездатність, правильне функціонування, дефект, несправність, контроль технічного стану, алгоритм технічного діагностування.

Розглянуті основи технічного та тестового діагностування. Проведений аналіз існуючих несправностей в цифрових системах.

Проведений огляд та аналіз існуючих систем тестування цифрових систем на кристалах. Зроблений висновок щодо недоліків даних систем, на основі яких базується мета магістерської роботи.

Метою магістерської роботи є дослідження методів та засобів вдосконалення алгоритмів пошуку дефектів у цифрових системах за допомогою дедуктивних та дедуктивно-паралельних методів та розроблення рекомендацій щодо застосування розглянутих методів.

РОЗДІЛ 2 КЛАСИФІКАЦІЯ ТА ПОБУДОВА АЛГОРИТМІВ ДІАГНОСТУВАННЯ

2.1 Основні визначення в алгоритмах діагностування

Елементарна перевірка (ЕП) – частина процесу діагностики, яка характеризується подачею на об'єкт тестовим або робочим впливом і отриманням з об'єкта відповіді.

Таблиця функцій несправностей (ТФН) – прямокутна таблиця, рядки якої є тестові набори, а стовпці - безліч всіх можливих технічних станів об'єкта діагностики; перший стовпець показує справну поведінку. На перетині рядків і стовпців ставляться реакції схеми на внесені несправності (рисунок 2.1).

R		E				
		e	...	e_i	...	$e_{[s]}$
A	π_1	R_1		R_1^i		$R_1^{[s]}$
	⋮					
	π_j	R_j		R_j^i		$R_j^{[s]}$
	⋮					
	$\pi_{[n]}$	$R_{[n]}$		$R_{[n]}^i$		$R_{[n]}^{[s]}$

Рисунок 2.1 – Таблиця функцій несправностей

E – множина технічних станів об'єкта діагностики.

e – справний стан об'єкта.

e_i – i - несправний стан об'єкта.

A – множина допустимих елементарних перевірок.

π – елементарна перевірка.

R – множина результатів елементарних перевірок.

R_j^i – результат елементарної перевірки об'єкта, що знаходиться в технічному стані e_i .

Перелік несправностей – на відміну від ТФН не має стовпців справної поведінки, а стовпці відповідають класам еквівалентних несправностей.

Константна несправність – постійне значення сигналу на лінії схеми, що виникає в результаті електричних або механічних дефектів елемента.

Класи еквівалентності – дефекти називаються еквівалентними, якщо при подачі будь-якого набору реакція на виходах схеми однакова або нема набору для розрізнення цих дефектів.

Довжина алгоритму діагнозу – середнє число елементарних перевірок, реалізація яких необхідна для виділення одного технічного стану.

Безпосереднє використання таблиці функцій несправностей як форми подання інформації при побудові і реалізації алгоритмів діагностики і фізичних моделей об'єктів часто неможливо через високу розмірності таблиці.

Однак як універсальна математична модель об'єкта діагностики таблиця функцій несправностей дуже наочна і зручна при обговоренні та класифікації принципів, а також основних процедур побудови та реалізації алгоритмів діагностики, навіть якщо ці принципи і процедури спочатку формулюються на мовах, відмінних від мови таблиць функцій несправностей.

Безліч π володіє властивістю виявлення будь-якої несправності з множини S , тобто для будь-якої несправності $s_i \in S$ знайдеться хоча б одна елементарна перевірка, що $R_J = R_J^i$, а також властивістю розрізнення всіх несправностей з множини S .

2.2 Властивості таблиць функцій несправностей

На рисунку 2.2 наведена схема, що складається з трьох елементів і відповідна їй таблиця несправностей

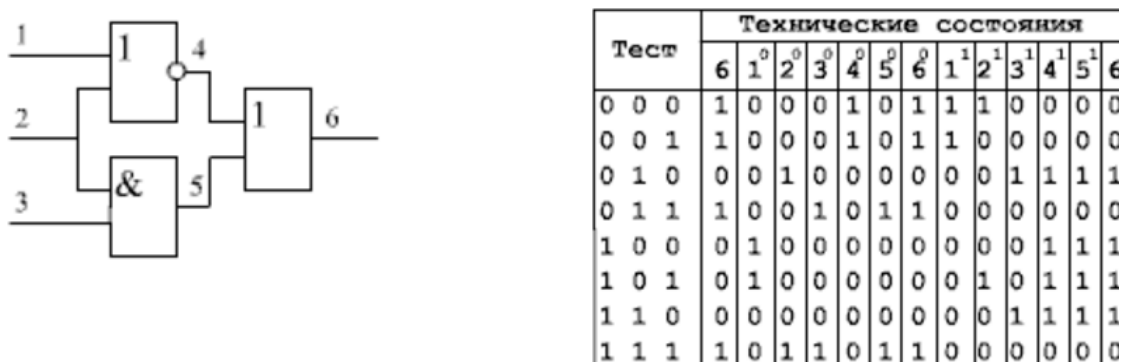


Рисунок 2.2 – Схема з таблицею несправностей

Тестом даної цифрової схеми розглядається вся множина двійкових наборів. Цифрами 1...6 позначені лінії схеми; стовпці ТН є технічними станами: наприклад стовпець 3⁰ означає

несправність логічного типу "постійний нуль на лінії 3". Стовець 6 відповідає реакції справної схеми

Для пошуку несправностей можна користуватися ТН. Після проведення діагностичного експерименту результати перевірки цифрового пристрою порівнюють послідовно з усіма стовпчиками ТН і при збігу визначають тип константної або іншого виду несправності.

Як уже зазначалося, для реальних цифрових схем ТН може мати значні розміри, що ускладнює їх застосування.

Однак існують способи, що дозволяють зменшити розміри ТН. Одним з них є спосіб, при якому з ТН видаляються надлишкові рядки і формується мінімальна таблиця.

ТН без надлишкових наборів характеризується одноразовим покриттям, тобто в кожному стовпці повинна стояти хоча б одна 1.

Рядок є не надлишковий, якщо при його видаленні з ТН порушується принцип одноразового покриття. Для даної ТН не надлишковими рядками є набори 000, 010, 011 і 101.

Як видно з рисунку 2.2, тестові набори задовольняють вимогам не надлишковості, так як вони покривають своїми одиницями всі стовбці ТН. Однакові стовбці ТН з нульовими векторами для кожного виду несправності відповідають несправностям, що не розрізняються тестом, що використовується. В наведеному прикладі ТП співпадають стовбці 3^0 та 5^0 , 4^1 , 5^1 та 6^1 .

На рисунках 2.3-2.5 наведено етапи перетворення ТН в скорочену таблицю.

Етап 1 – Визначаємо не надлишкові рядки

Етап 2 – Видаляємо надлишкові рядки.

Етап 3 – Будуємо скорочену таблицю переходів.

Тест	Технические состояния												
	6	1 ⁰	2 ⁰	3 ⁰	4 ⁰	5 ⁰	6 ⁰	1 ¹	2 ¹	3 ¹	4 ¹	5 ¹	6 ¹
0 0 0	1	0	0	0	1	0	1	1	1	0	0	0	0
0 0 1	1	0	0	0	1	0	1	1	0	0	0	0	0
0 1 0	0	0	1	0	0	0	0	0	0	1	1	1	1
0 1 1	1	0	0	1	0	1	1	0	0	0	0	0	0
1 0 0	0	1	0	0	0	0	0	0	0	0	1	1	1
1 0 1	0	1	0	0	0	0	0	0	1	0	1	1	1
1 1 0	0	0	0	0	0	0	0	0	0	1	1	1	1
1 1 1	1	0	1	1	0	1	1	0	0	0	0	0	0

Рисунок 2.3 – Перший етап перетворення таблиці несправності в скорочену таблицю

Тест	Технические состояния													
	6	1 ⁰	2 ⁰	3 ⁰	4 ⁰	5 ⁰	6 ⁰	1 ¹	2 ¹	3 ¹	4 ¹	5 ¹	6 ¹	
0 0 0	1	0	0	0	1	0	1	1	1	0	0	0	0	
0 1 0	0	0	1	0	0	0	0	0	0	1	1	1	1	
0 1 1	1	0	0	1	0	1	1	0	0	0	0	0	0	
1 0 1	0	1	0	0	0	0	0	0	1	0	1	1	1	

Рисунок 2.4 – Другий етап перетворення таблиці несправності в скорочену таблицю

Тест	Технические состояния													
	6	1 ⁰	2 ⁰	3 ⁰	4 ⁰	5 ⁰	6 ⁰	1 ¹	2 ¹	3 ¹	4 ¹	5 ¹	6 ¹	
0 0 0	1	0	0	0	1	0	1	1	1	0	0	0	0	
0 1 0	0	0	1	0	0	0	0	0	0	1	1	1	1	
0 1 1	1	0	0	1	0	1	1	0	0	0	0	0	0	
1 0 1	0	1	0	0	0	0	0	0	1	0	1	1	1	

Рисунок 2.5 – Третій етап перетворення таблиці несправності в скорочену таблицю

2.3 Побудова умовного алгоритму пошуку дефектів у вигляді дерева

Умовний алгоритм пошуку дефектів у вигляді дерева (надалі дерево пошуку дефектів) будується на основі ТН, для якої дотримуються всі три умови – виявлення, розрізнення і мінімальність. В даному алгоритмі пошуку дефекту в якості елементарної перевірки P_i розглядається подача на схему одного двійкового набору і порівняння реакцій схеми в контрольних точках (на зовнішніх виходах) з еталоном. Результат елементарної перевірки вважається позитивним, якщо отримане значення сигналу на зовнішньому виході не співпало з еталоном, тобто тестовий набір виявив дефект в схемі. Результат елементарної перевірки вважається негативним, якщо значення на зовнішньому виході збіглося з еталоном, тобто тестовий набір не виявив дефект в схемі. Елементарна перевірка P_i розбиває всю множину еквівалентних дефектів на дві підмножини: виявляються (перевіряються) цією перевіркою (1 у відповідних стовбцях) і ті, які не перевіряються (0 або пробіл у відповідних стовбцях в залежності від форми ТН).

В умовному алгоритмі пошуку дефектів порядок проходження елементарних перевірок залежить від обраних критеріїв оптимальності і результатів попередніх перевірок.

При побудові алгоритму діагностування і відповідного йому діагностичного дерева найбільш часто використовують критерій мінімальних середніх витрат, що витрачаються на отримання результатів діагностики. Однак на практиці використовують алгоритми близькі до оптимальних. Такі алгоритми, і відповідні їм діагностичні дерева, можуть бути отримані шляхом включення в діагностичне дерево на кожному кроці перевірки значення деякої функції переваги f , яке мінімізує або максимізує значення. В якості такої функції переваги може виступати функція, яка ґрунтується на кількості інформації, одержаної при проведенні відповідної перевірки. Одним з інформаційних критеріїв для обчислення функції переваги є критерій половинного ділення, тобто чергова елементарна перевірка повинна розбивати підмножину підозрюваних дефектів навпіл (на дві приблизно рівних підмножини). Отримане за такою функції переваги дерево буде близьким до симетричного, тобто буде мінімізована максимальна довжина гілки (кількість перевірок для знаходження будь-якого дефекту). Нижче наведена формула обчислення даної функції переваги на кожному кроці побудови дерева пошуку дефектів.

$$F = \left| \sum R^0 - \sum R^1 \right| \rightarrow \min, \quad (2.1)$$

де R^0 - порожні (нульові) значення в розглянутій рядку ТН,

R^1 - поодинокі значення в розглянутій рядку ТН.

Таким чином, максимальному значенню функції переваги відповідає рівність кількості одиничних і порожніх (нульових) значень в розглянутому рядку ТН.

Такий вибір функції переваги дозволяє зробити алгоритм діагностування більш симетричним, тобто всі гілки діагностичного дерева, що представляє цей алгоритм, будуть мати приблизно однакову довжину. Це означає, що для виявлення кожної несправності в схемі при проведенні діагностичного експерименту потрібно буде приблизно однакове число кроків, тобто такий алгоритм має середню мінімальну довжину.

Початкова множина підозрюваних класів еквівалентних дефектів (КЕД) схеми, що має один вихід, містить всі КЕД, тобто $K_1, K_2, K_3, K_4, K_5, K_6, K_7$. Перелік елементарних перевірок і функція переваги f для першого кроку побудови дерева пошуку дефектів показані на рисунку 2.6.

		K_1	K_2	K_3	K_4	K_5	K_6	K_7	f
P_1	0	1	1	0	1				5
P_2	1	1	1	0				1	3
P_3	0	0	1	0		1	1	1	1
P_4	0	1	0	0		1	1	1	1
P_5	0	1	1	1				1	3

Рисунок 2.6 – Перелік елементарних перевірок і функція переваги f для першого кроку побудови дерева пошуку дефектів

За функції переваги першою елементарною перевіркою можна вибрати P_4 , яка розбиває всю безліч КЕД на два підмножини, які перевіряються цією перевіркою - K_5, K_6, K_7 і не перевіряються нею - K_1, K_2, K_3, K_4 .

На рисунку 2.7 наведено результуюче дерево пошуку дефектів, побудоване на підставі ТН, яка розглядається, для виділених класів еквівалентних дефектів.

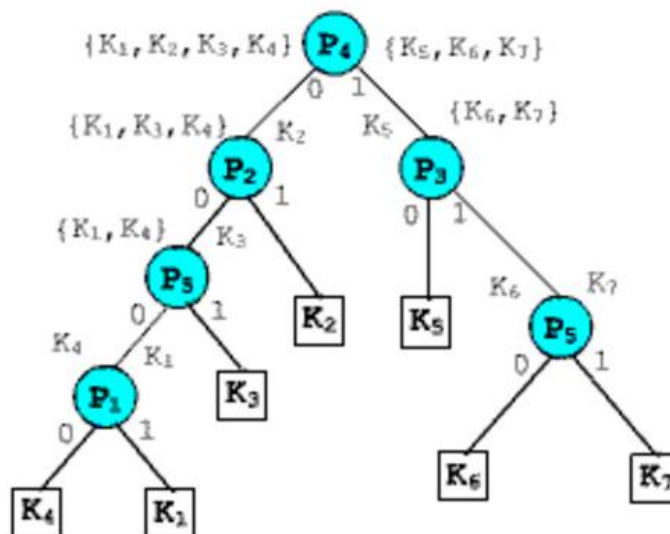


Рисунок 2.7 – Результуюче дерево пошуку дефектів

Процес пошуку довільного дефекту в схемі з використанням отриманого дерева пошуку дефектів називається діагностичним експериментом.

2.4 Алгоритми діагностики

Алгоритм діагностики задає сукупність елементарних перевірок, послідовність (або послідовності) їх реалізації та правила обробки результатів, що реалізуються елементарними перевірками з метою отримання результатів діагностики.

Сукупність елементарних перевірок, які входять до алгоритму діагностики позначається символом T . Для позначення будь-якої елементарної перевірки з сукупності T використовується символ t_j .

Для більш наочного уявлення класифікації алгоритмів діагнозу і характерних цільових функцій їх оптимізації розглянемо деякі питання подання алгоритмів діагнозу у вигляді графів.

Результати будь-якої елементарної перевірки можуть бути використані як ознаки розбиття множини E технічних станів об'єкта або підмножин цієї множини на класи. Використовуючи таке трактування елементарних перевірок і їх результатів, будь-який алгоритм діагнозу можна уявити деяким орієнтованим графом. В даному випадку граф, що представляє алгоритм діагностики, є деревом (рисунок 2.8).

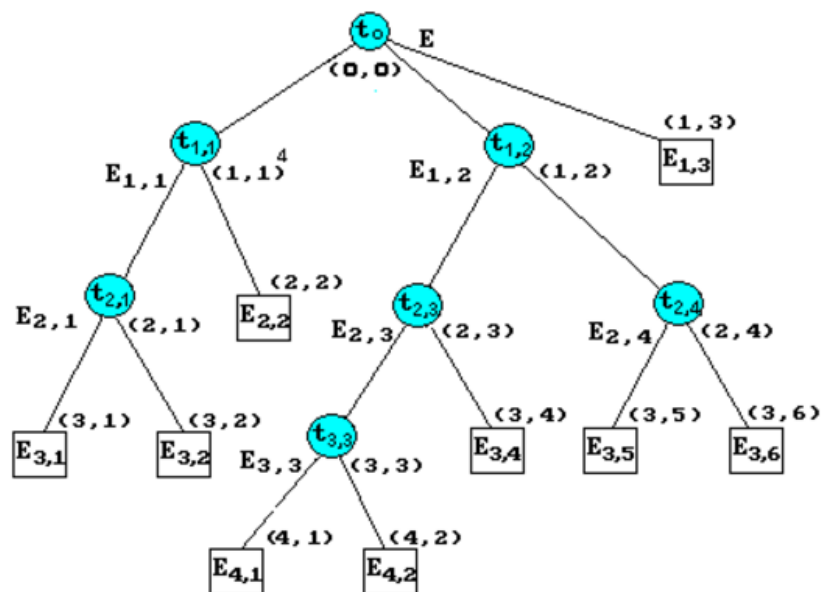


Рисунок 2.8 – Приклад дерева, що представляє алгоритм діагностики

Дерево має вершини двох типів: вершини, з яких виходить хоча б одна дуга, і вершини, з яких не виходить жодної дуги.

На рисунку 2.8 представлені вершини двох типів – першого типу з затемненими кружками, а вершини другого типу – світлими кружками.

У дереві є єдина вершина першого типу, в яку не входить жодна дуга. Ця вершина називається початковою або коренем дерева (вершина відзначена символами t_0 , E).

Вершини, з яких не виходить жодної дуги, називаються звичайною або висячою вершинами. Решта вершини дерева називаються внутрішніми.

У кожен вершину дерева, крім його кореня, заходить тільки одна дуга.

Рангом вершини дерева називається число дуг шляху, який починається в початковій вершині t_0 і закінчується в розглянутій вершині.

Мінімальним рангом P_0 дерева є ранг, для якого існує хоча б одна висяча вершина і не існує жодної висячої вершини рангу менше P_0 .

Для даного прикладу мінімальний ранг дерева дорівнює 1 (перше число в дужках для вершин дерева E1.1 ... E1.3)

Максимальним рангом P_m дерева є ранг, для якого існує хоча б одна вершина рангу P_m і не існує жодної вершини для рангу більше P_m .

Для даного прикладу максимальний ранг дерева дорівнює 4 (перше число в дужках для вершин E4.1 і E4.2)

Внутрішні і висячі вершини дерева нумеруються парою чисел в дужках (p, d) , де p є ранг вершини, а d – її порядковий номер серед вершин одного і того ж рангу (на прикладі, зліва направо в заданому графічному поданні дерева).

Корінь дерева позначається парою чисел в дужках $(0,0)$.

Початковою і внутрішнім вершин дерева зіставляються елементарні перевірки множини T дуг дерева, що виходить з деякою (початкової або внутрішньої) його вершини, зіставляються можливі результати перевірки, що подається цією вершиною. Крім того, початковій вершині дерева ставиться у відповідність множина E можливих технічних станів об'єкта, а внутрішнім і висячим вершин підмножини технічних станів, отримані як класи розбиття за результатами відповідних елементарних перевірок.

Елементарну перевірку і підмножину технічних станів, що зіставлені вершині дерева, позначається символами tp, d і E_p, d відповідно. Безліч елементарних перевірок tp, d , зіставлених початковій і всім внутрішнім вершин дерева, позначається символом T_d .

2.5 Умовний алгоритм пошуку дефектів

Для проведення діагностичного експерименту необхідно мати реальний цифровий пристрій, який підозрюється на наявність дефекту і алгоритм пошуку дефекту у вигляді дерева.

Контрольною точкою (КТ) у розглянутому діагностичному експерименті виступає зовнішній вихід схеми, яка представлена на рисунку 2.9 та її дерево.

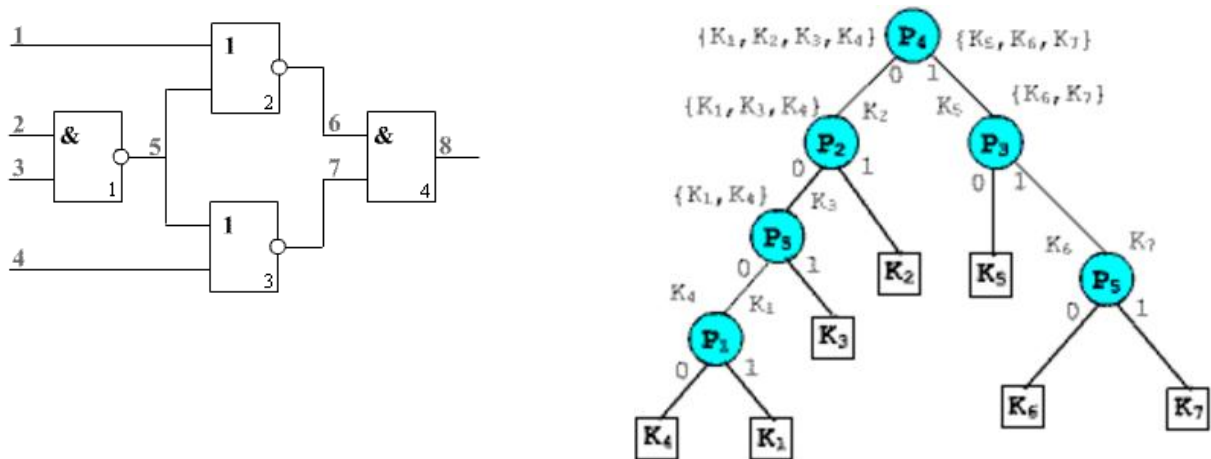


Рисунок 2.9 – Схема для діагностичного експерименту та її дерево

У якості ЕП розглядається подача на входи цифрової схеми конкретного двійкового набору P_i (зазначеного в черговий вершині дерева пошуку дефекту) і порівняння отриманого значення сигналу в контрольній точці з еталоном (результатом справного моделювання).

Результат ЕП вважається позитивним, якщо отримане значення сигналу на виході схеми не співпало з еталоном, тобто тестовий набір виявив несправність.

Результат ЕП вважається негативним якщо отримане значення сигналу на виході схеми збіглося з еталоном, тобто тестовий набір не виявив несправність.

Припустимо, що на лінії 5 даної схеми є дефект у вигляді постійного значення 0 (константа 0) (рисунок 2.10).

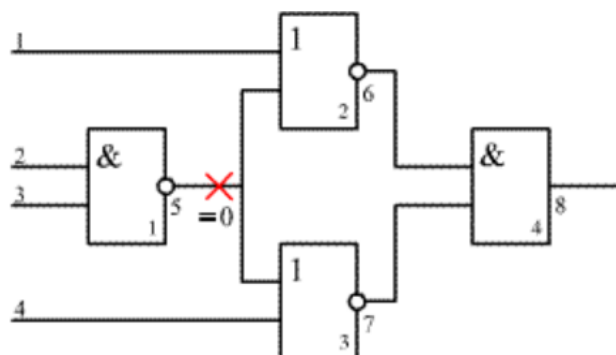
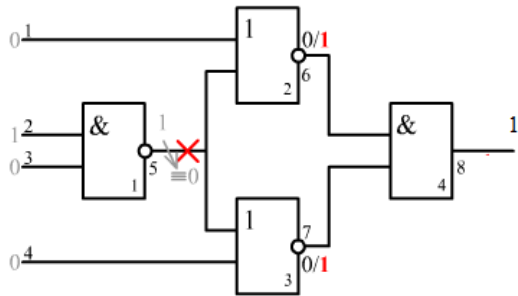


Рисунок 2.10 – Дефект схеми на лінії 5

Згідно побудованого раніше дерева пошуку дефектів подамо на зовнішні входи схеми відповідні виконавчі набори. Насамперед подамо згідно дерева пошуку дефектів набір, що відповідає вершині P4 (0 1 0 0) (рисунок 2.11).



На виході отримали «1» замість «0».

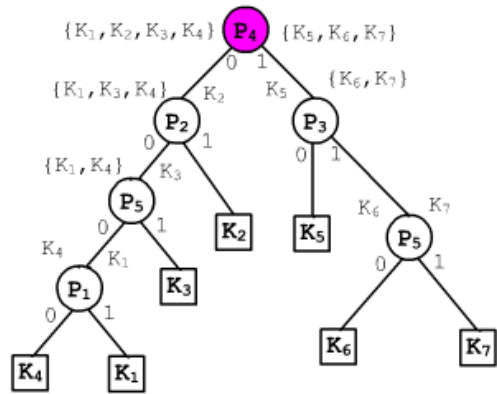
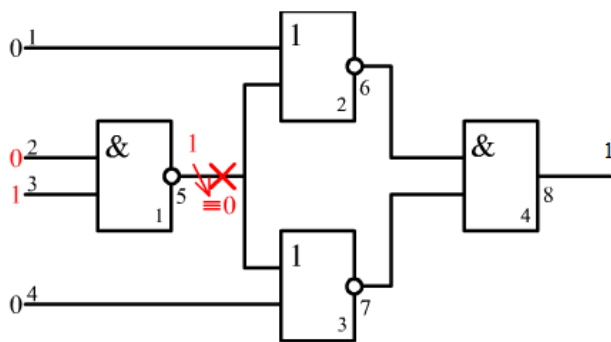


Рисунок 2.11 – Схема для діагностичного експерименту з несправністю та її дерево

Так як результат перевірки позитивний (значення на виході не співпало з еталонним), рухаємося по правій гілці дерева і в якості наступної перевірки вибираємо P3 (0 0 1 0) (рисунок 2.12).



На виході отримали «1» замість «0».

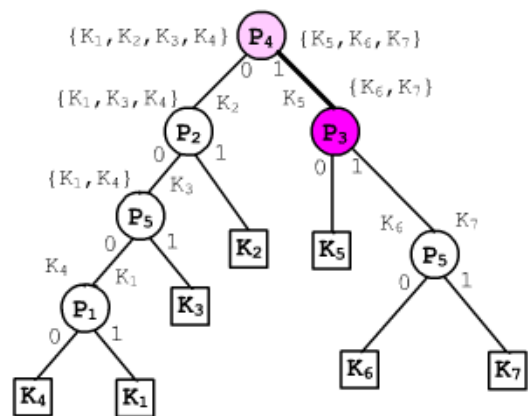
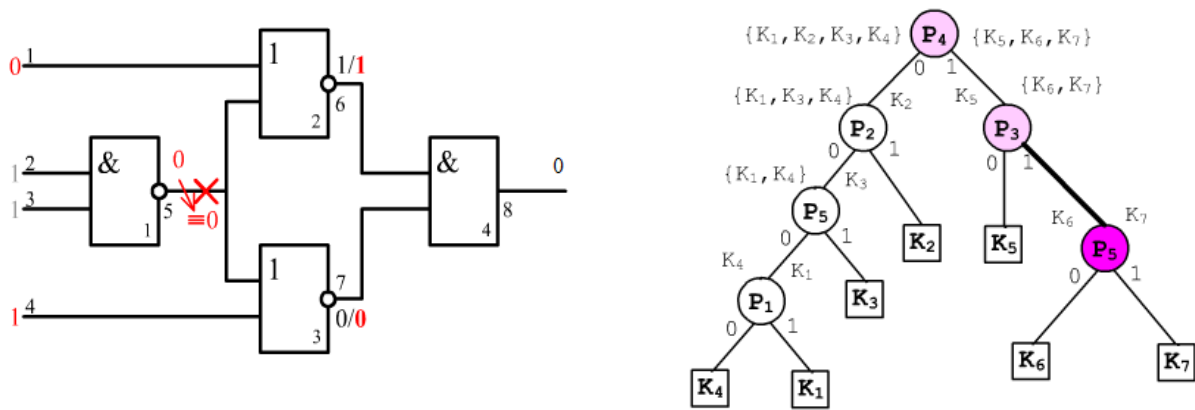


Рисунок 2.12 – Схема для діагностичного експерименту з несправністю та її дерево

Результат цієї перевірки теж позитивний, значення на виході не співпало з еталонним, тому рухаємося по правій гілці дерева і в якості наступної перевірки вибираємо P5 (0 1 1 1) (рисунок 2.13).



На виході отримали «0».

Рисунок 2.13 – Схема для діагностичного експерименту з несправністю та її дерево

Результат на виході схеми збігся з еталонним (перевірка негативна), отже по лівій гілці дерева вибираємо термінальну вершину, відповідного класу еквівалентних дефектів K_6 . Повернувшись до вихідного списку класів еквівалентних дефектів бачимо, що в цьому класі знаходиться один дефект 5^0 .

2.6 Висновки до розділу 2

У другому розділі магістерської роботи розглянуто класифікацію та побудову алгоритмів діагностування, що використовують таблиці несправностей. Визначені такі поняття, як перелік несправностей, константна несправність, класи еквівалентності та довжина алгоритму діагнозу. Зазначено, що безпосереднє використання таблиці функцій несправностей, як форми подання інформації при побудові і реалізації алгоритмів діагностики і фізичних моделей об'єктів, часто неможливо через високу розмірність таблиці.

Однак як універсальна математична модель об'єкта діагностики таблиця функцій несправностей дуже наочна і зручна при обговоренні та класифікації принципів, а також основних процедур побудови та реалізації алгоритмів діагностики, навіть якщо ці принципи і процедури спочатку формулюються на мовах, відмінних від мови таблиць функцій несправностей. Визначені властивості таблиць функцій несправності. Запропоновано використання умовного алгоритму пошуку дефектів у вигляді дерева, яке будується на основі таблиці несправності, для якої дотримуються три умови - виявлення, розрізнення і мінімальність. В даному алгоритмі пошуку дефекту в якості елементарної перевірки P_i розглядається подача на схему одного двійкового набору і порівняння реакцій схеми в контрольних точках (на зовнішніх виходах) з еталоном.

РОЗДІЛ 3 СТРУКТУРНІ АЛГОРИТМИ ПОШУКУ ДЕФЕКТІВ В СИСТЕМАХ ДІАГНОСТУВАННЯ

3.1 Основні визначення в структурних алгоритмах пошуку дефектів

Тестом називаються вхідні впливи на схему та еталонні реакції на дані впливи в справному стані. Еталонні реакції на тестові впливи згорнуті в еталонні сигнатури. Передбачається, що тест побудований таким чином, що всі константні несправності, що виникають в схемі, транспортуються до зовнішніх (спостережуваних) входів.

Сигнатура – 4-х розрядний шістнадцятковий код, який є зверткою двійкової послідовності за допомогою сигнатурного регістра з лінійними зворотними зв'язками.

Об'єкт діагностування (ОД) – це об'єкт, над яким проводиться експеримент елементарної перевірки. ОД є цифрові схеми без глобальних зворотних зв'язків (ЗЗ) з фізично доступними внутрішніми лініями, кожна з яких може виступати в якості контрольної точки. Для даної схеми існує тест Т, що перевіряє всі дефекти схеми, і здійснює їх транспортування до зовнішніх виходів схеми. Еталонні реакції на даний тест представлені у вигляді набору еталонних сигнатур $S_{ет}$ у всіх КТ (на всіх лініях схеми).

ЕП характеризується вхідними впливами, поданими на ОД, з яких знімаються реакції ОД на ці впливи, а також еталонні реакції в КТ на вхідні впливи.

Результатом перевірки є збіг або розбіжність еталонних реакцій КТ.

Зондова елементарна перевірка – це ЕП, що проводиться з використанням механічного (гальванічного) зонду у внутрішніх контрольних точках схеми, не пов'язаних із зовнішнім роз'ємом. Зонд пов'язаний з сигнатурним аналізатором, який згортає експериментальну двійкову послідовність в контрольній точці в сигнатуру, доступну для подальшого аналізу.

Алгоритм пошуку дефектів (алгоритм діагностування) – це сукупність ЕП, послідовність їх реалізації, а також правила обробки їх результатів з метою проведення процесу діагностування.

Контрольна точка – це фізично доступна лінія схеми або роз'єму цифрового пристрою з якої відбувається зняття еталонних реакцій при подачі на схему тестових впливів.

Попередники для обраної лінії цифровий схеми список ліній і примітивів, інформація від яких може пройти до даної лінії в штатному режимі роботи схеми.

Наступники – для обраної лінії цифровий схеми список ліній і примітивів, інформація до яких може пройти від даної лінії в штатному режимі роботи схеми. Область підозрюваних дефектів – перелік контрольних точок, які аналізуються на кожному наступному кроці алгоритму пошуку дефектів і серед яких знаходиться лінія, підозрювана на наявність дефекту.

Діагностичний експеримент (ДЕ) над схемою здійснюється в два етапи. На першому етапі проводиться безумовний експеримент шляхом подачі на схему тесту Т і порівняння двійкових послідовностей на зовнішніх виходах схеми з еталонними. Якщо результат хоча б на одному виході не збігається з еталоном, виконується другий етап діагностичного експерименту з використанням зондового пошуку дефектів. У початкову область підозрюваних дефектів входять всі попередники контактів роз'єму, що не співпали. Для одновиходової схеми (підсхеми) в область підозрюваних дефектів входять всі лінії схеми. Глибина пошуку дефекту – до еквіпотенційної лінії схеми.

В ході реалізації умовного діагностичного експерименту реалізується зондовий алгоритм пошуку дефекту. В якості ЕП використовується подача на ОД тесту Т і порівняння в потрібній точці контролю експериментальних сигнатур $S_{\text{екс}}$ з еталонними $S_{\text{ет}}$. Якщо $S_{\text{екс}}$ збігається з $S_{\text{ет}}$, результат перевірки вважається позитивним, а якщо не збігається – негативним. КТ вибираються таким чином, щоб вони знаходилися поза циклами локальних ОС. В якості КТ не використовуються зовнішні виходи (контакти роз'єму), так як вони перевірялися в ході першого етапу ДЕ. Вартості виконання всіх елементарних перевірок рівні між собою.

ДЕ проводиться виходячи з припущення, що в схемі є одиночний дефект, і після його знаходження відбувається відновлення працездатності схеми. В умовах наявності кратного дефекту відновлення працездатності виконується по ітеративній процедурі: після усунення чергового дефекту ДЕ повторюється знову і так до тих пір, поки при проведенні безумовного ДЕ всі експериментальні значення виходів не співпадуть з еталонними.

Основний принцип, що лежить в основі зондового алгоритму пошуку дефектів наступний. Якщо в черговій КТ результат ЕП негативний, то в область підозрюваних дефектів на кожному наступному кроці алгоритму входить сама лінія КТ і всі її попередники. Якщо результат перевірки позитивний, то всі попередники передбачаються справними, а підозрювана дефектна лінія знаходиться серед інших КТ області підозрюваних дефектів попереднього кроку алгоритму.

Умовний діагностичний експеримент будується з умовою мінімізації максимальної послідовності елементарних перевірок, що реалізується із застосуванням стратегії половинного ділення.

3.2 Методи стиску двійкових послідовностей. Сигнатурний аналіз

При діагностиці сучасних цифрових пристроїв виникає необхідність порівняння виконавчих послідовностей великої довжини на виходах пристрою, що перевіряються, з еталонними двійковими послідовностями. Найбільш доцільно відображати виконавчі послідовності в більш компактний простір перевірки.

Сигнатурний аналіз є методом, в основі якого лежить стиск інформації. Він перетворює двійкові послідовності будь-якої довжини в певному вузлі схеми в сигнатуру з набору чотирьох шістнадцятиричних цифр, що однозначно характеризує цей вузол.

Його застосування для діагностування засноване на тому принципі, що справна цифрова схема при періодичному порушенні одного і того ж входу буде завжди видавати однаковий вихідний сигнал, перетворений в сигнатуру. Якщо ж цей періодичний вихідний сигнал відрізняється від еталонного, то схема несправна.

Дана методика дозволяє з високою точністю локалізувати несправність і не вимагає високої кваліфікації обслуговуючого персоналу.

Математичною основою сигнатурного аналізу є спосіб кодування двійкових послідовностей з використанням циклічних кодів. При цьому будь-яке двійкове число описується багаточленом, що містить фіктивну змінну X .

Вихідній двійковій послідовності з певного вузла цифрового пристрою відповідає поліном $G(X)$ ступеня $n-1$, де n - число розрядів двійкового коду. У процесі формування поліном $G(X)$ ділиться на породжуючий поліном $P(X)$, значення якого визначається структурою регістра зсуву з зворотними зв'язками в сигнатурному аналізаторі. Для формування чотирирозрядної шістнадцятиричної сигнатури зі всієї безлічі можливих поліномів вибирають $P(X) = 1 + X^6 + X^8 + X^{11} + X^{15}$, що відповідає зворотним зв'язкам від 7, 9, 12 і 16 розрядів.

При діленні $G(X)$ на $P(X)$ отримуємо частку $Q(X)$ і залишок $R(X)$. Вихідний поліном при цьому дорівнює $G(X) = P(X)Q(X) \oplus R(X)$, де \oplus – знак суми за модулем 2.

Математична модель сигнатурного аналізатора (розподіл полінома на поліном) представлена на прикладах на рисунках 3.1 та 3.2.

Наприклад, в нас є:

$$\text{Ділене} - S(X) = X^6 - 2X^5 - X^4 + 2X^3 - X + 1$$

$$\text{Дільник} - G(X) = X^3 - 2X^2 + 3X - 1$$

$$\text{Частка} - Q(X)$$

$$\text{Залишок} - R(X)$$

$$\begin{array}{r|l}
 -X^6 - 2X^5 - X^4 + 2X^3 - X + 1 & X^3 - 2X^2 + 3X - 1 \\
 \hline
 X^6 - 2X^5 + 3X^4 - X^3 & X^3 - 4X - 5 \\
 \hline
 -4X^4 + 3X^3 - X + 1 & \\
 -4X^4 + 8X^3 - 12X^2 + 4X & \\
 \hline
 & -5X^3 + 12X^2 - 5X + 1 \\
 & -5X^3 + 10X^2 - 15X + 5 \\
 \hline
 & R(X) \leftarrow 2X^2 + 10X - 4
 \end{array}$$

Рисунок 3.1 – Ділення поліному

Двійковий поліном – двійковий код. Коефіцієнт – 0,1, знаки +, ступені – номер розряду.
 Так як поліном – не позиційний код, то сума за модулем 2 дорівнює відніманню.

Наприклад:

Код – 1101011011

Поліном – $S(X) = X^9 + X^8 + X^6 + X^4 + X^3 + X + 1$

Утворений поліном – $G(X) = X^3 + X + 1$

$$\begin{array}{r|l}
 \oplus X^9 + X^8 + X^6 + X^4 + X^3 + X + 1 & X^3 + X + 1 \\
 X^9 + X^7 + X^6 & \hline
 X^6 + X^5 + X^4 + X^3 + X & \\
 \hline
 \oplus X^8 + X^7 + X^4 + X^3 + X + 1 & \\
 X^8 + X^6 + X^5 & \\
 \hline
 \oplus X^7 + X^6 + X^5 + X^4 + X^3 + X + 1 & \\
 X^7 + X^5 + X^4 & \\
 \hline
 \oplus X^6 + X^3 + X + 1 & \\
 X^6 + X^4 + X^3 & \\
 \hline
 \oplus X^4 + X + 1 & \\
 X^4 + X^2 + X & \\
 \hline
 X^2 + 1 &
 \end{array}$$

Рисунок 3.2 – Сума за модулем 2 поліному

Частка дорівнює $Q(X) = 1111010$, а залишок – $R(X) = 101$.

Двійкове ділення – зсув вправо.

Ділення на поліном – регістр зі зворотними зв'язками.

Розглянемо процес формування сигнатур. Вміст 16-ти розрядного регістра, відповідне залишку $R(X)$, відображається в шестнадцятиричному форматі. На рисунках 3.3 – 3.5 показаний процес формування сигнатури для 16-ти розрядної послідовності $X = 1111110000011111$. На вхід суматора по модулю 2 подається двійковий код X і сигнали зворотного зв'язку з відповідних виходів тригерів регістра зсуву. У початковому стані регістр містить всі нулі. Потім на кожному часовому такті на вхід суматора по модулю 2 надходять виконавчі сигнали вхідного коду. Перші сім тактів регістра суматор працює як зсувний регістр без включення зворотних зв'язків. Після 7-го такту одиниця з 7-го тригера регістра надходить на вхід суматора, складається з нульовим значенням X на восьмому такті і на вході суматора по модулю 2 з'являється 1, яка надходить на вхід першого розряду регістра. Аналогічно формуються сигнали на вході регістру для інших тимчасових тактів. Після шістнадцяти тактів отримаємо шістнадцяткове число – сигнатуру, яке відповідає бінарному коду.

Сигнал на зворотні зв'язки йде з затримкою в один такт.

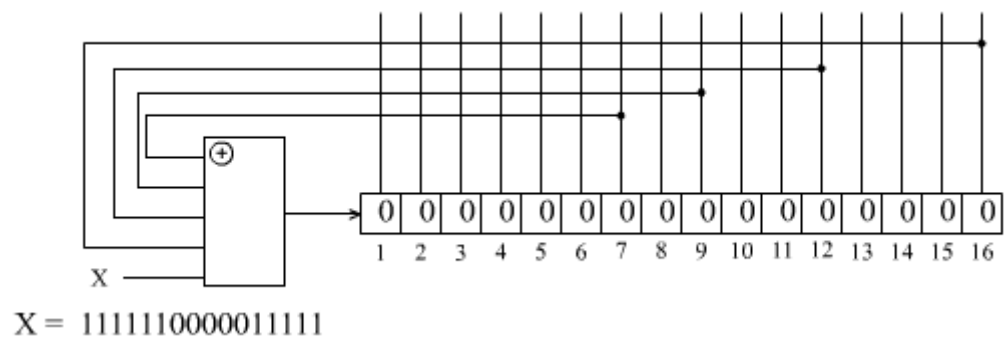


Рисунок 3.3 – Процес формування сигнатури для 16-ти розрядної послідовності. Перший етап

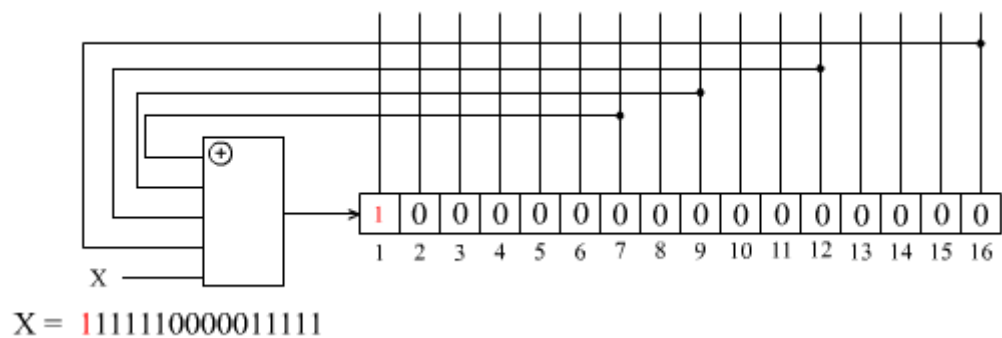


Рисунок 3.4 – Процес формування сигнатури для 16-ти розрядної послідовності. Другий етап

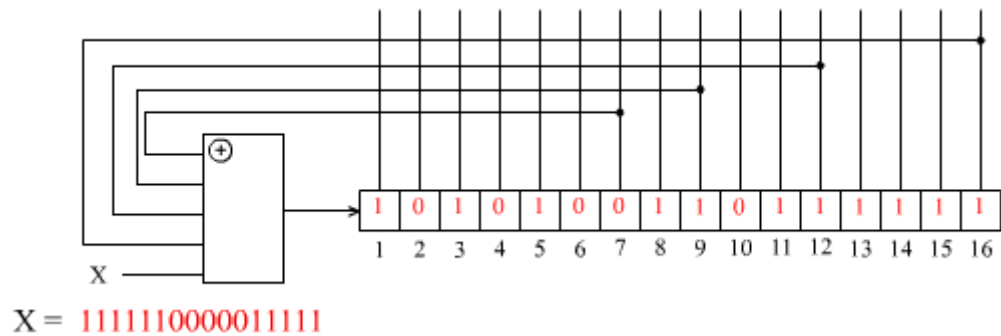


Рисунок 3.5 – Процес формування сигнатури для 16-ти розрядної послідовності.

Останній етап

На останньому етапі процес формування сигнатури для 16-ти розрядної послідовності $X = 1111110000011111$ в шістнадцятковій системі має такий вигляд: A9BF.

3.3 Зондовий алгоритм пошуку дефектів. Зворотнє простеження

Алгоритм діагностування в даному випадку не диференціюється чітко на процедури контролю та пошуку дефектів. Використання сигнатурного аналізу визначає ЕП як отримання експериментальної сигнатурної реакції на довільному спостерігаємому контактї схеми при подачі на неї повного тесту. Перевірка вважається позитивною, якщо на потрібній точці контролю еталонна сигнатура, обчислена за відсутності несправностей у схемі, збігається з експериментальної. В іншому випадку перевірка буде негативною. Точкою контролю є зовнішній контакт компонента цифрового приладами (мікросхеми, роз'єми типового елемента заміни (ТЕЗ), роз'єму всього пристрою).

Два логічних результати кожної елементарної перевірки задають альтернативні шляхи пошуку дефекту або визначення технічного стану ОД. У цьому сенсі діагностування є умовним і зондовим, оскільки має місце зняття сигнатур з використанням логічного зонду і сигнатурного логічного аналізатора. При кожному застосуванні такого алгоритму реалізується одна з можливих послідовностей діагностування залежно від технічного стану ОД і завдяки вибору чергової елементарної перевірки за результатами попередніх. Тривалість і вартість виконання умовного діагностичного експерименту вище, ніж у безумовного. До того ж він вимагає значних витрат на створення "інтелектуального" алгоритму пошуку дефектів. Метод половинного ділення як відома модифікація підходу до мінімізації числа елементарних перевірок може бути віднесений до умовних алгоритмів, коли чергова точка контролю розбиває

безліч несправностей на два приблизно однакових підмножини. Результат перевірки в черговій крапці контролю зменшує підозрювану область наявності дефектів вдвічі.

При безумовному експерименті розпізнавання технічного стану на безлічі можливих, але наперед заданих станів, здійснюється тільки з аналізу реакцій зовнішніх виходів на тест, що накладає більш жорсткі вимоги до останнього, який повинен бути діагностуючим. В іншому випадку технічний стан об'єкта може бути не ідентифіковано. Крім того, для реальних схем обсяг діагностичної інформації (ДІ) може перевищувати межі доцільності її застосування, що робить завжди актуальною задачу мінімізації ДІ. Одним з ефективності та технологічних підходів вирішення згаданої проблеми є метод сигнатурного аналізу. Його математична основа - отримання залишку (сигнатури) від ділення як завгодно довгої кінцевої двійкової послідовності на певне двійкове число (утворює поліном). Практичне розв'язання такого завдання зводиться до використання регістра зсуву з зворотними зв'язками з розрядів 7, 9, 12, 16, які заведені разом з лінією X, що є входом сигнатурного аналізатора, на суматор по модулю два. Такий регістр прийнято називати сигнатурним регістром.

Після подачі двійкової послідовності на вхід X, отриманий код-сигнатура станів розрядів регістра з ймовірністю $P = 0,9998$ відображає вихідний вхідний вектор довільної довжини. Для зручності 16-розрядна двійкова сигнатура записується в шістнадцятковий алфавіт (0, 1, ..., 9, A, B, C, D, E, F) чотирма символами, які слугують "паспортом" для кожного контакту мікросхеми або зовнішніх контактів роз'єму при виконанні тестового діагностування. Сигнатура контакту, отримана у вигляді реакції на заданий тест без наявності в схемі несправностей, називається еталонною. Якщо в реальному об'єкті експериментальна сигнатура на контакті не дорівнює еталонній, то розглянутий контакт або його попередник підозрюються несправними. В іншому випадку (при рівності сигнатур) підграф попередників визнається справним. Кількість повних циклів подачі тесту для пошуку дефекту з використанням сигнатурного аналізу в гіршому випадку може дорівнювати числу контрольних точок для зняття зондом аналізатора експериментальних сигнатур, розташованих на логічному шляху від несправного контакту до зовнішніх пристроїв. Для зменшення часу циклу подача тесту здійснюється на частотах, близьких до робітників для реального цифрового пристрою (ЦП). Відсутність мінімізації кількості контрольних точок в пропонованому далі граф-методі пошуку дефектів, які використовують ідею зворотного простежування несправності за структурою об'єкта, пов'язане з введенням класом макродефектів, які розпізнаються при послідовному виконанні перевірок ліній логічного шляху.

Діагностична модель пристрою представлена графом функціонально-гальванічних зв'язків контактів мікросхем і роз'ємів цифрового пристрою. Для побудови алгоритму діагностування необхідні: повний перевіряючий тест щодо одиночних константних

несправностей, еталонні сигнатури всіх зовнішніх контактів мікросхем і роз'ємів, структурна схема ЦП. З метою наближення моделі несправностей до реальних дефектів, введемо такі типи стану об'єкта діагностування (фізичних дефектів):

$$F = \{F_0, F_1, F_2, F_3, F_4, F_5, F_6\}, \quad (3.1)$$

де F_0 – справний стан об'єкта;

F_1 – відсутність гальванічного зв'язку між контактами різних елементів мікросхем;

F_2 – несправність на виході елемента;

F_3 – відсутність зв'язку між входом і шиною константи 0,1;

F_4 – несправність зовнішнього вхідного контакту ОД;

F_5 – несправність виходів об'єднаних в монтажну логіку;

F_6 – несправність, що циркулює в контурі глобальної зворотного зв'язку.

Всі можливі точки контролю (фізично доступні контакти) ідентифікуються порядковими номерами або топологічними ідентифікаторами (координата мікросхеми, номер ніжки). Послідовність діагностування контактів на топології схеми визначається двома напрямками: праворуч – ліворуч – при негативній перевірці; зверху – вниз – при позитивній. Така стратегія задає як чергові точки контролю при негативній перевірці на виході елемента його верхній вхідний контакт. При фіксації позитивного результату на вхідному контакті примітиву чергова точка контролю задається розташованою нижче вхідної ніжки. Якщо в останньому випадку перевірка негативна, чергова точка контролю визначається вихідним контактом елемента, пов'язаного з даним входом. Якщо в діагностичному експерименті розглядається чергова точка контролю, що не є вихідним контактом схеми, то всі попередні перевірки ліній – наступників мали негативний результат. Для запропонованого прикладу результат побудови алгоритму пошуку дефектів представлений в таблиці 6.1. Припустимо, що першою точкою контролю є контакт 16. Він має еталонну сигнатуру $S = 9F3A$. (Всі контакти також мають власні, в даному випадку фіктивні, сигнатури). Схема побудови алгоритму представлена на рисунку 3.6.

При реалізації алгоритму зворотного простежування зняття сигнатур починається з зовнішніх виходів, а конкретно з будь-якого з зовнішніх виходів, де експериментальна сигнатура не збіглася з еталонною.

Таблиця 3.1 – Результат побудови алгоритму

КТ	П+	П-	sign
16	17	08	9F3A
17	18	14	7A32
18	F0	15	4F98
14	F1	10	7A32
15	F1	12	4F98
10	11	F3	A3FA
11	F2	09	FAF1
12	13	09	FAF1
13	F2	04	4A2F
08	F1	05	9F3A
09	F1	05	FAF1
05	06	01	9FFC
06	07	01	3A76
07	F2	03	F83A
01	F1	F4	9FFC
02	F1	F4	3A76
03	F1	F4	F83A
04	F1	F4	4A2F

Результат порівняння еталонної і експериментальної сигнатур дозволяє вибрати чергові точки контролю на наступному кроці діагностування при двох альтернативах:

П+ – перевірка позитивна (другий стовпець таблиці);

П- – перевірка негативна (третій стовпець таблиці).

Стан F0 фіксується, якщо всі виходи дають позитивну перевірку; F1 – на гальванічного зв'язку зафіксована позитивна перевірка при негативній попередньої;

F2 – при негативній перевірці на виході всі входи мають позитивний результат порівняння сигнатур; F3 – негативна перевірка на вході, пов'язаної з шинами 0 і 1; F4 – негативна перевірка на зовнішньому вході цифрового пристрою.

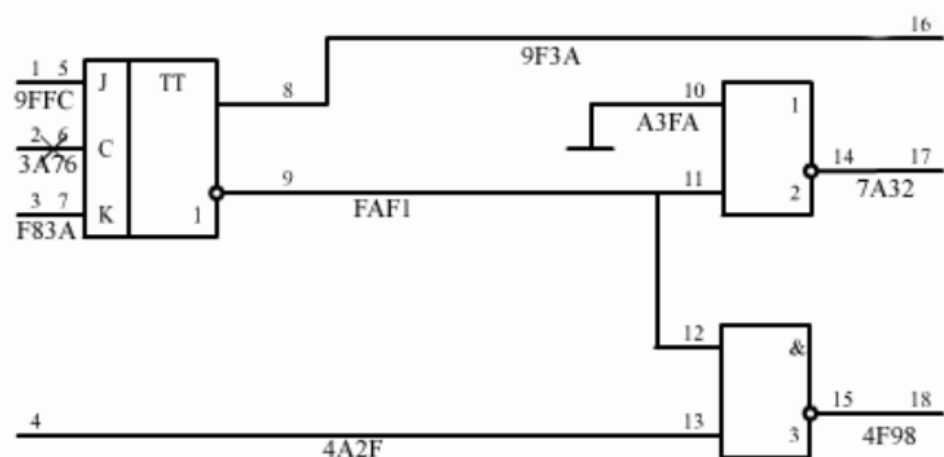


Рисунок 3.6 – Схема побудови алгоритму пошуку дефектів

Таблиця графа пошуку дефектів має число рядків, що дорівнює кількості спостережуваних контактів і дозволяє здійснювати діагностичний експеримент в діалоговому режимі без наявності принципової електричної схеми об'єкта.

Функції оператора полягають в ініціалізації команд для подачі тесту, установки зонду сигнатурного аналізатора в координату точки контролю, висвічується на дисплеї, прийнятті остаточного рішення про стан об'єкта.

Для проведення діагностичного експерименту припустимо, що в схемі є розрив зв'язку між лініями 2 та 6. Перевірка початкової координати 16 дає негативний результат, що визначає черговою точкою контролю контакт 8. Його перевірка має від'ємне значення, отже, переходимо на координату 5. Вона дає позитивне випробування, а значить виконується перехід до лінії 6. На ній фіксується негативний результат порівняння сигнатур, що визначає перехід до точки 2, в якій перевірка виявилася позитивною. Висновок: технічний стан ОД - F1 – немає зв'язку між контактами 2 і 6. Пример реалізації пошуку даного дефекту представлений на рисунках 3.7 – 3.11.

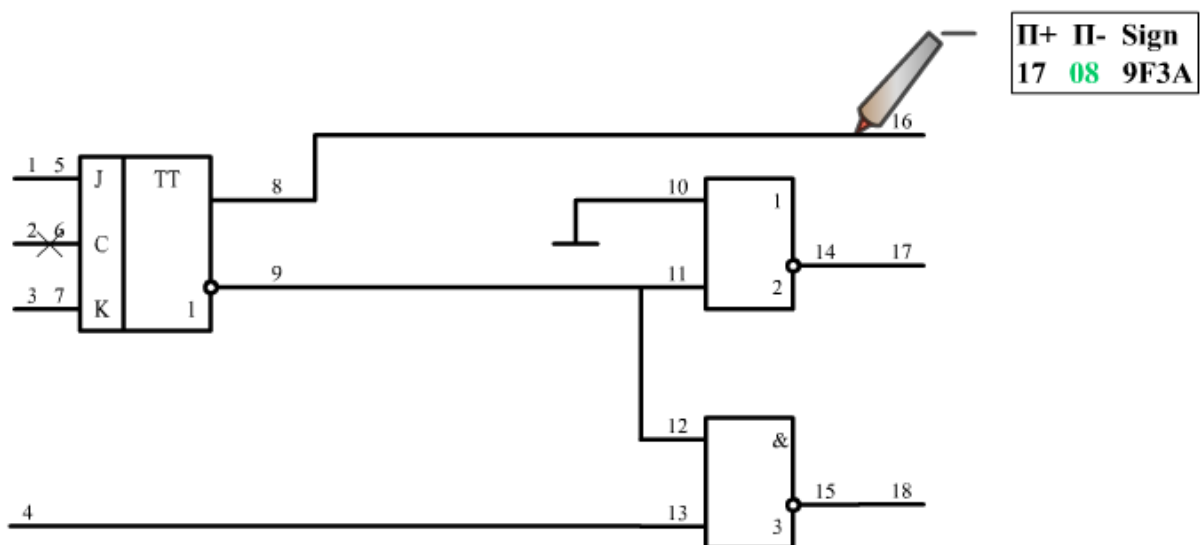


Рисунок 3.7 – Реалізація пошуку дефекту. Крок перший

Для зменшення кількості точок контролю попередньо його повинен виконувати структурний аналіз схеми щодо елементарних перевірок зовнішніх виходів. У припущенні наявності в схемі одиночного макродефекту, що характерно для об'єкта в період експлуатації, несправність слід шукати в області, отриманої перетином підсхем, що відносяться до тих спостережуваних виходів, на яких зафіксовані негативні перевірки. З підозрюваної області слід виключити ті точки контролю, які мають логічні шляхи до виходів з позитивним результатом

випробування. При допущенні пошуку кратних дефектів область існування несправностей дорівнює асоціації підсхем, логічно пов'язаних з справними виходами.

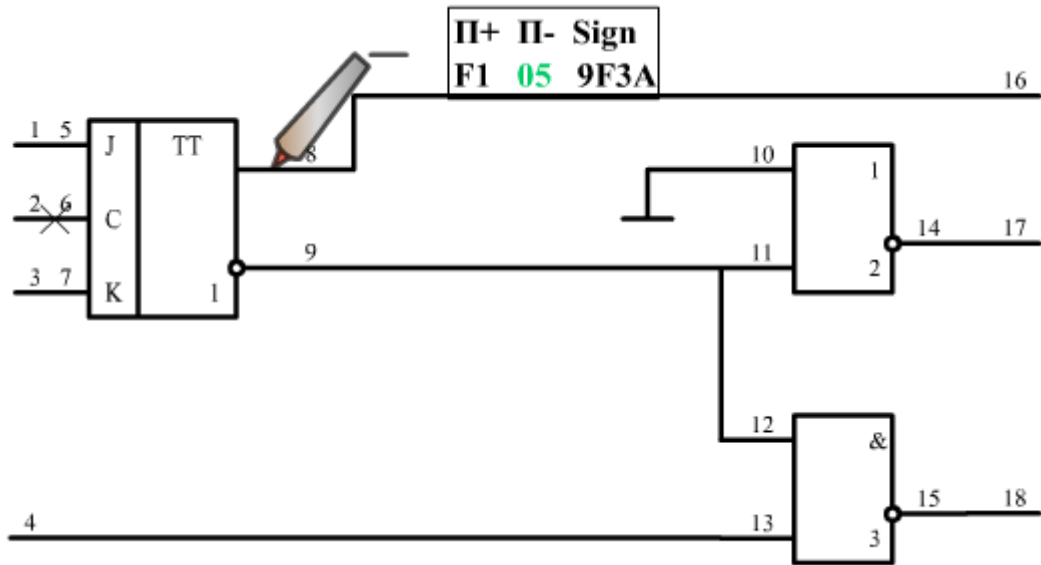


Рисунок 3.8 – Реалізація пошуку дефекту. Крок другий

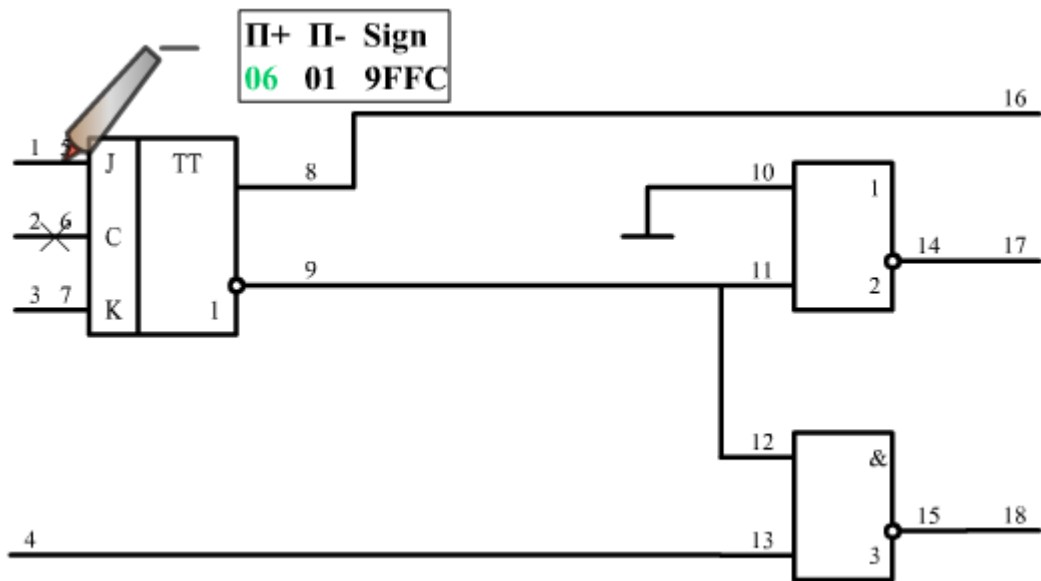


Рисунок 3.9 – Реалізація пошуку дефекту. Крок третій

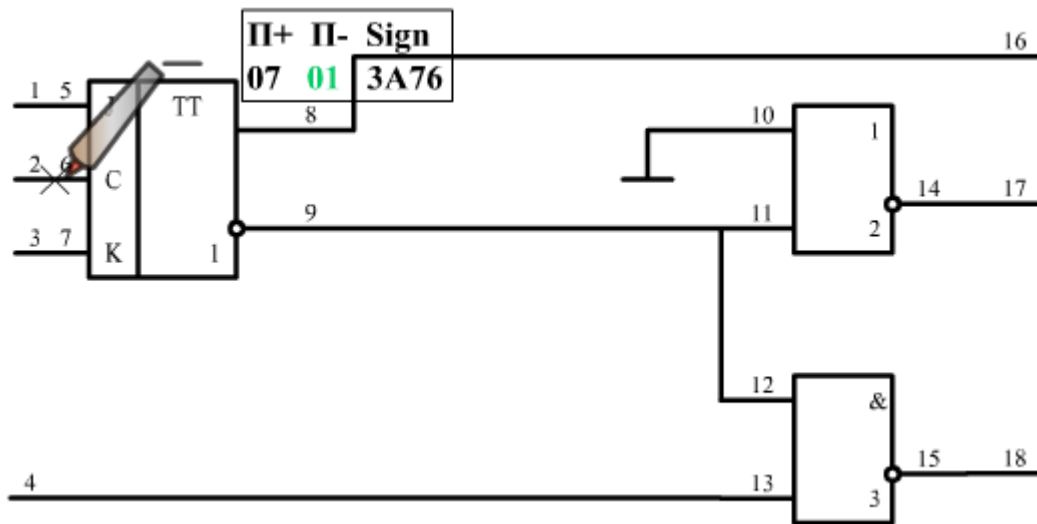


Рисунок 3.10 – Реалізація пошуку дефекту. Крок четвертий

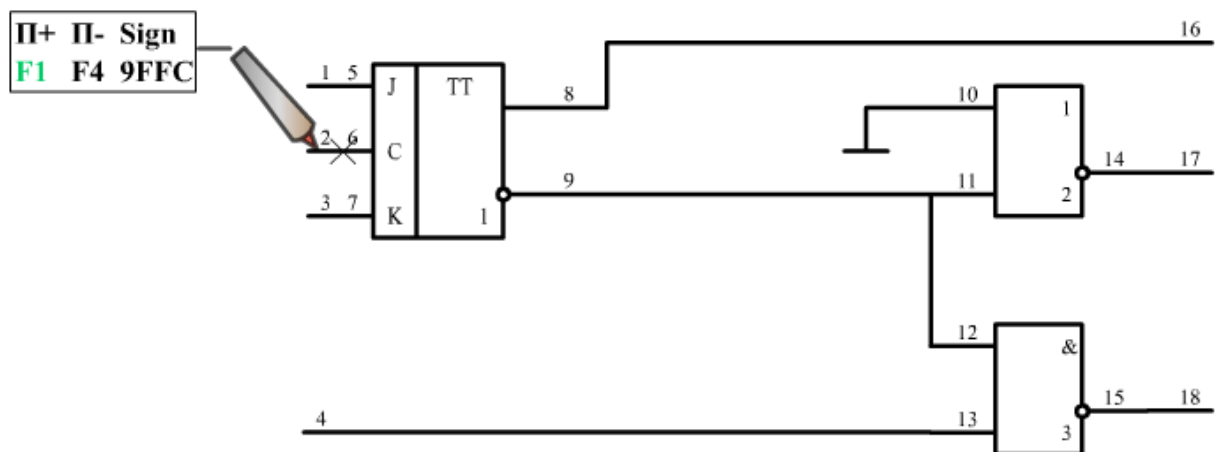


Рисунок 3.11 – Реалізація пошуку дефекту. Крок п'ятий

3.4 Умовний зондовий алгоритм пошуку дефекту

Наприклад, дана структурна модель схеми (рисунок 3.12), для якої передбачається побудова алгоритму пошуку дефектів. Кожен ПЕ описується системою підграфів взаємозв'язку його входів-виходів. Для одновиходових ПЕ підграф один.

Алгоритм складається з наступних кроків:

- 1) Ранжування схеми.
- 2) Побудова графа.
- 3) Побудова матриці досяжності.

4) Побудова алгоритму пошуку дефекту.

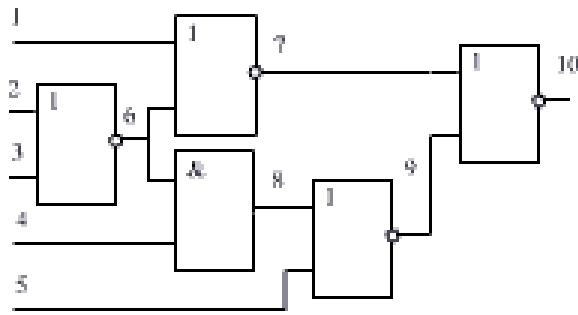


Рисунок 3.12 – Структурна модель схеми

Ранжування схеми виконується шляхом нумерації її ліній.

1) Виконується нумерація по порядку зовнішніх входів схеми, тобто ліній не мають попередників.

2) Наступними по порядку номерами нумеруються виходи примітивів, входи яких вже пронумеровані, і які не є зовнішніми виходами, тобто мають наступників.

3) Пункт 2 виконується до тих пір, поки не виявляться занумерованих всі внутрішні лінії схеми.

4) Наступними по порядку номерами нумеруються зовнішні виходи схем, тобто лінії не мають спадкоємців.

На рисунку 3.12 зображена проранжована схема.

На підставі нумерованої схеми будується граф (рисунок 3.13), вершинами якого є нумеровані лінії, а лінії графа визначають гальванічний (електричний) зв'язок між ними.

Для прикладу розглянемо вершину 6. Є дуги між вершиною 6 і вершинами 2,3,7,8.

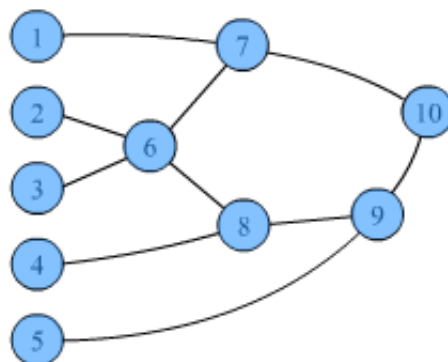


Рисунок 3.13 – Граф, побудований на основі схеми

Зі схеми видно, що лінія 6 гальванічно пов'язана з лініями 2, 3 (через елемент "АБО-НІ") і з лініями 8 і 7 (через елементи "І" і "АБО-НІ" відповідно

Для отриманого графа будується матриця досяжності (таблиця 6.2), рядки якої відповідають попередникам, а стовпці приймачам деякої лінії. У даній матриці одиницями заповнюється головна діагональ і ті клітини в рядках, які є попередниками для лінії головної діагоналі. При цьому на лініях зовнішніх входів "1" коштує тільки на головній діагоналі (немає попередників), а для зовнішнього виходу одновиходової підсхеми – все одиниці.

Для одновиходової схеми побудова умовного алгоритму пошуку дефектів починається в припущенні, що в підозрювану безліч дефектів входять всі лінії схеми. Для багатовиходної схеми початкова множина підозрюваних дефектів визначається за результатами проведення першого етапу діагностичного експерименту.

Побудова дерева пошуку дефектів виконується за ітеративною процедурою, на кожному кроці якої розглядається своя підмножина підозрюваних дефектів.

Для поточної підмножини підозрюваних дефектів будується матриця досяжності і обчислюється функція переваги стратегії половинного ділення за формулою:

$$f = \min | D_i \& M_j - D_i / 2 |, \quad (3.2)$$

де D_i – Поточна безліч підозрюваних дефектів, M_j – j -й рядок матриці досяжності.

Таблиця 3.2 – Матриця досяжності на першому кроці алгоритму

М	1	2	3	4	5	6	7	8	9	10	f
1	1										4
2		1									4
3			1								4
4				1							4
5					1						4
6		1	1			1					2
7	1	1	1			1	1				0
8		1	1	1		1		1			0
9		1	1	1	1	1		1	1		2
10	1	1	1	1	1	1	1	1	1	1	5

Вибирається чергова перевірка з мінімальним значенням функції переваги.

Обрана перевірка розбиває безліч підозрюваних дефектів на два підмножини:

- за негативним результатом перевірки (сигнатура не збігається) – попередники і сама лінія, що перевіряється;
- за позитивним результатом перевірки (сигнатура збігається) – інші лінії початкового безлічі підозрюваних дефектів.

Попередні два пункти повторюються до тих пір, поки в підозрюваних підмножинах не опиняться по одній лінії (термінальні вершини).

Нижче наведено приклад побудови дерева з використанням алгоритму вище.

На першому кроці, використовуючи формулу (3.2) вирахуємо функцію переваги стратегії половинного ділення: $D_i/2 = 10/2 = 5$. Вибирається чергова перевірка з мінімальним значенням функції переваги. $f = \min(0)$ для 2-х ліній схеми: 7 і 8. Це говорить про те, що в якості першої точки перевірки можна вибрати як лінію 7, так і лінію 8. Обираємо лінію 8 (тому що вона знаходиться ближче до виходу 10, на якому зафіксовано розбіжність з еталоном на першому етапі ДЕ). Тоді дерево пошуку дефектів приймає вид, зображений на рисунку 3.14.

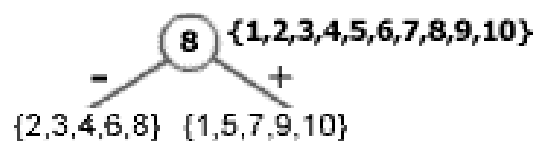


Рисунок 3.14 – Дерево пошуку дефектів після першого кроку

Перевірка сигнатури на лінії 8 розділяє безліч підозрюваних дефектів на 2 підмножини:

- За негативною перевіркою: {2,3,4,6,8} (Попередники і сама перевіряема лінія).
- За позитивною перевіркою: {1,5,7,9,10} (Всі інші).

У матриці досяжності попередники і перевіряється лінія виділені кольором.

Переходимо до наступного кроку: з підмножини дефектів, що підозрюються, {1,5,7,9,10} необхідно вибрати таку точку перевірки.

Наступну точку перевірки було вибрано з підмножини {1,5,7,9,10}, а не з підмножини {2,3,4,6,8}, бо в ході побудови дерева пошуку дефектів перебираються всі варіанти, і до другої підмножини в ході алгоритму доведеться ще повернутися.

Перед переходом до наступних кроків алгоритму треба зазначити що:

- у формулі обчислення функції переваги D_i & M_j на кожному кроці використовується одна і та ж матриця досяжності, а змінюється тільки підмножина підозрюваних дефектів D_i . Це зручно для комп'ютерної програми побудови дерева. При побудові дерева вручну доцільно для наочності на кожному кроці алгоритму будувати фрагмент матриці досяжності, пов'язаний з

поточною підмножиною підозрюваних дефектів D_i , що буде видно при виконанні наступних кроків алгоритму.

На другому кроці для поточної підмножини підозрюваних дефектів $\{1,5,7,9,10\}$ будується матриця досяжності (таблиця 3.3), і знову обчислюється функція переваги стратегії половинного ділення за формулою (3.2), де $D_i = 5/2 = 2.5$.

Таблиця 3.3 – Матриця досяжності на другому кроці алгоритму

M	1	5	7	9	10	f
1	1					1,5
5		1				1,5
7	1		1			0,5
9		1		1		0,5
10	1	1	1	1	1	2,5

Вибирається чергова перевірка з мінімальним значенням функції переваги.

$f = \min$ для двох ліній 7 і 9. Можна вибрати будь-яку точку в якості точки чергової перевірки, тому була обрана лінія 9. Перевірка в цій точці розбиває безліч підозрюваних дефектів на 2 підмножини:

- За негативною сигнатурою: $\{5,9\}$ (Попередники і сама перевіряема лінія).
- За позитивною сигнатурою: $\{1,7,10\}$ (Всі інші).

Тоді дерево пошуку дефектів приймає вид, зображений на рисунку 3.15.

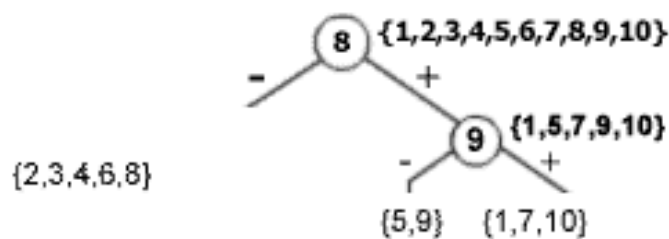


Рисунок 3.15 – Дерево пошуку дефектів після другого кроку

Виконується перехід до наступного кроку: з підмножини $\{1,7,10\}$ необхідно вибрати наступну точку перевірки.

Для поточної підмножини підозрюваних дефектів $\{1,7,10\}$ будується матриця досяжності (таблиця 3.4) і знову обчислюється функція переваги стратегії половинного ділення за формулою (3.2), де $D_i = 3/2 = 1.5$.

Таблиця 3.4 – Матриця досяжності на третьому кроці алгоритму

M	1	7	10	f
1	1			0,5
7	1	1	1	0,5
10	1	1	1	1,5

Вибирається чергова перевірка з мінімальним значенням функції переваги. $f = \min$ для двох ліній 1 і 7. В якості точки чергової перевірки вибирається лінія 7. Перевірка в цій точці розбиває безліч підозрюваних дефектів на 2 підмножини:

- За негативною сигнатурою: $\{1,7\}$ (Попередники і сама перевіряема лінія).
- За позитивною сигнатурою: $\{10\}$ (Всі інші).

В одному з отриманих множин є тільки 1 елемент $\{10\}$. Це означає, що в разі перевірки схеми на лініях 8, потім 9, і потім 7 і збіги сигнатур з еталонними в усіх точках несправна лінія 10. При цьому слід пам'ятати, що сигнатура в точці 10 свідомо не збігається з еталонною, тобто в схемі присутня несправність.

Якщо хоча б в одній точці сигнатура не збігається з еталонною, то потрібні додаткові перевірки, про що свідчать незавершені гілки дерева пошуку дефектів.

Тоді дерево пошуку дефектів приймає вид, зображений на рисунку 3.16.

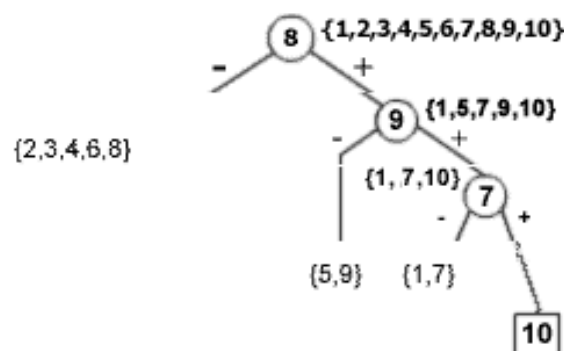


Рисунок 3.16 – Дерево пошуку дефектів після третього кроку

Виконується перехід до наступного кроку: з підмножини $\{1,7\}$ необхідно вибрати наступну точку перевірки.

На четвертому кроці для поточної підмножини підозрюваних дефектів $\{1,7\}$ будувати матрицю не треба, бо в точці 7 перевірка вже проводилась на другому кроці в поточній гілці, і єдина точка, де необхідно провести перевірку - це лінія 1.

В результаті перевірки сигнатури на лінії 1 однозначно буде виявлено місце дефекту:

- за негативною сигнатурою: точка виникнення несправності – лінія 1.
- за позитивною сигнатурою: точка виникнення несправності – лінія 7.

Матриця досяжності на цьому кроці зображена в таблиці 3.5, дерево досяжності – на рисунку 3.17.

Таблиця 3.5 – Матриця досяжності на четвертому кроці алгоритму

M	1	7	f
1	1		0,5
7	1	1	0,5

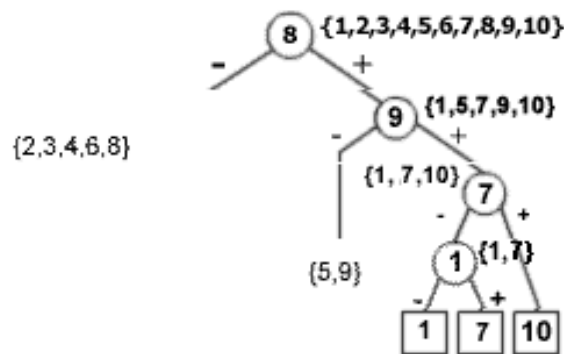


Рисунок 3.17 – Дерево пошуку дефектів після четвертого кроку

Виконується перехід до наступного кроку: виконується повернення до вершини 9 і підмножини можливих несправностей $\{5,9\}$

На п'ятому кроці для поточної підмножини підозрюваних дефектів $\{5,9\}$ не треба будувати матриці, бо в точці 9 перевірка вже проводилась на третьому кроці, тому єдиною точкою, де необхідно провести перевірку є лінія 5.

В результаті перевірки сигнатури на лінії 5 однозначно буде визначено місце дефекту:

- за негативною сигнатурою: точка виникнення несправності – лінія 5.
- за позитивною сигнатурою: точка виникнення несправності – лінія 9.

Тоді дерево пошуку дефектів приймає вид, зображений на рисунку 3.18.

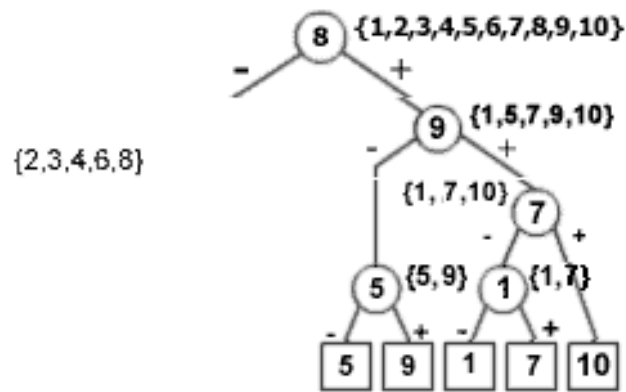


Рисунок 3.18 – Дерево пошуку дефектів після п'ятого кроку

Виконується перехід до наступного кроку: виконується повернення до кореня дерева – вершини 8 і підмножини можливих несправностей $\{2,3,4,6,8\}$.

На шостому кроці для поточної підмножини підозрюваних дефектів $\{2,3,4,6,8\}$ будується матриця досяжності (таблиця 3.6) і обчислюється функція переваги стратегії половинного ділення за формулою (3.2), де $D_i = 5/2 = 2,5$.

Таблиця 3.6 – Матриця досяжності на шостому кроці алгоритму

M	2	3	4	6	8	f
2	1					1,5
3		1				1,5
4			1			1,5
6	1	1		1		0,5
8	1	1	1	1	1	2,5

Вибирається чергова перевірка з мінімальним значенням функції переваги. $f = \min$ для 6 лінії схеми. Перевірка сигнатури на лінії 6 розділяє безліч підозрюваних дефектів на 2 підмножини:

- за негативною сигнатурою: $\{2,3,6\}$ (Попередники і сама лінія, що перевіряється);
- за позитивною сигнатурою: $\{4,8\}$ (Всі інші).

Тоді дерево пошуку дефектів приймає вид, зображений на рисунку 3.19.

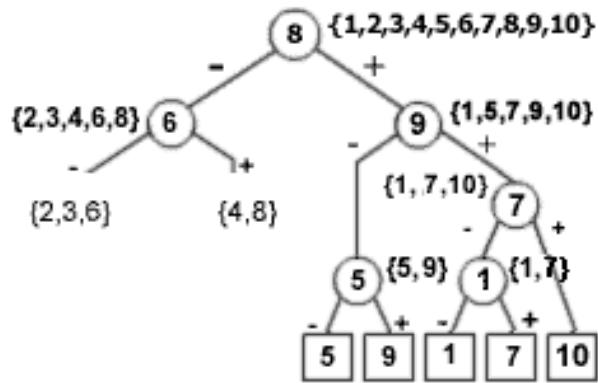


Рисунок 3.19 – Дерево пошуку дефектів після шостого кроку

Виконується перехід до наступного кроку: необхідно перевірити лінію 4.

В результаті перевірки сигнатури на лінії 5 однозначно буде визначено місце дефекту:

– за негативною сигнатурою: точка виникнення несправності – лінія 4;

– за позитивною сигнатурою: точка виникнення несправності – лінія 8.

Тоді дерево пошуку дефектів приймає вид, зображений на рисунку 3.20.

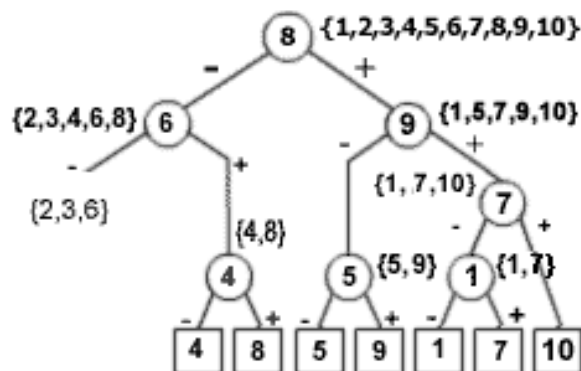


Рисунок 3.20 – Дерево пошуку дефектів після сьомого кроку

Виконується перехід до наступного кроку: виконується повернення до кореня дерева – вершини 8 і підмножини можливих несправностей $\{2,3,4,6,8\}$.

На восьмому кроці для поточної підмножини підозрюваних дефектів $\{2,3,6\}$ будується матриця досяжності (таблиця 3.7) і обчислюється функція переваги стратегії половинного ділення за формулою (3.2), де $D_i = 3/2 = 1,5$.

Таблиця 3.7 – Матриця досяжності на восьмому кроці алгоритму

M	2	3	f
2	1		1,5
3		1	1,5

Вибирається чергова перевірка з мінімальним значенням функції переваги. В якості точки перевірки вибирається лінія 2. Перевірка сигнатури на лінії 2 розділяє безліч підозрюваних дефектів на 2 підмножини:

- за негативною сигнатурою: {2} (Попередники і сама лінія, що перевіряється);
- за позитивною сигнатурою: {3,6} (Всі інші).

Виконується перехід до дев'ятого (останнього) кроку: виконується перевірка лінії 3.

За негативною сигнатурою: {3} (Попередники і сама перевіряється лінія).

За позитивною сигнатурою: {6} (Всі інші).

Дерево пошуку несправностей повністю побудоване і зображене на рисунку 3.21.

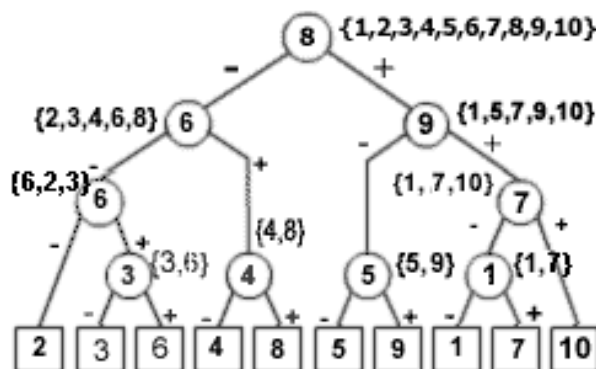


Рисунок 3.21 – Побудоване дерево пошуку несправностей

Нижче наведено приклад проведеного експерименту – внесення в схему одиночної несправності та її виявлення.

Умовний алгоритм пошуку дефекту зображений на рисунку 3.21.

Дано реальний цифровий пристрій з переліком фізично доступних контрольних точок. У розглянутій схемі (рисунок 3.22) контрольними точками є всі лінії схеми. У реальних пристроях може бути інакше. Засобом діагностування є ведений зонд, сполучений з сигнатурним аналізатором (СА).

Діагностичний експеримент проводиться в два етапи. На першому етапі після подачі на схему тесту значення сигналів на всіх зовнішніх виходах схеми порівнюються з еталонними.

Отримуємо на виході помилку, бо інакше немає сенсу в перевірці схеми, якщо вона і так вже працює. Виконується перехід до другого етапу діагностичного експерименту, в ході якого реалізується умовний алгоритм пошуку дефектів з використанням веденого зонду.

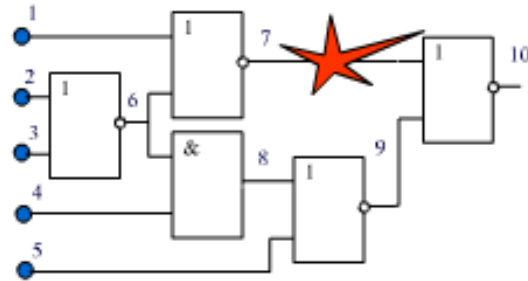


Рисунок 3.22 – Схема з внесеним дефектом

Зонд – це проста голка, котрою можна знімати сигнали з ліній схеми, проколюючи лак на доріжках і подаючи їх на СА (рисунок 3.23).



Рисунок 3.23 – Ведений зонд, з'єднаний з сигнатурним аналізатором

Є умовний алгоритм пошуку дефектів, що задає послідовність елементарних перевірок. (одновиходової схеми алгоритм пошуку дефекту може будуватися до початку проведення ДЕ, а для багатовиходової схеми в оперативному режимі після проведення першого етапу ДЕ). В якості елементарної перевірки використовується подача на схему тесту і порівняння сигнатури в обраній КТ з еталонною.

Припустимо в схемі є дефект на лінії 7, який треба виявити. Подаємо на схему тесті порівнюємо на виході отримане значення з еталонним. Результат на виході схеми не збігається з еталоном, тому виконується перехід до другого етапу ДЕ і виконується пошук дефекту за допомогою зонду (рисунок 3.24).

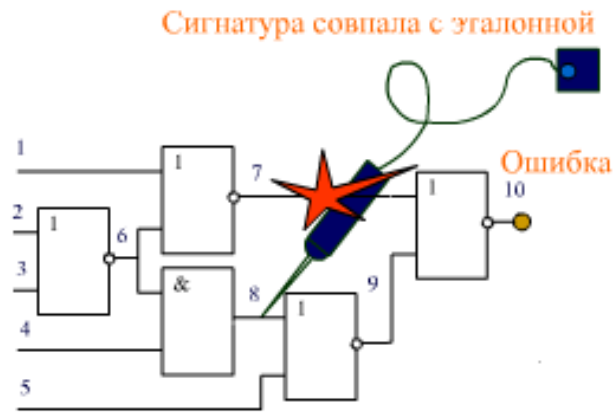


Рисунок 3.24 – Пошук дефекту за допомогою зонду

Кореневої вершиною дерева несправностей є елементарна перевірка на лінії 8. Це означає, що треба ткнути зондом на лінію 8 і подати на схему тест.

Порівнюємо отриману сигнатуру з еталонною.

Сигнатура співпаде, тобто перевірка дає позитивний результат.

Дивимося на дерево (пошук несправностей) і вибираємо позитивну (праву) гілку, що виходить з вершини 8. Такий вибір відразу виключає зі списку підозрюваних дефектів лінії 2, 3, 4, 6. Їх перевіряти вже не треба, так як вони є попередниками справної лінії 8. Чи іншими словами, будь-яка несправність на лініях 2, 3, 4, 6 неминуче і невідворотно викликала би й несправність на лінії 8. Так як лінія 8 в нормі, то і лінії 2, 3, 4, 6 вважаються робочими. Ситуацію, коли виникають помилки на двох лініях одночасно і при якихось умовах вони нейтралізують один одного не розглядаються в зв'язку з малою вірогідністю.

Наступна елементарна перевірка – лінія 9. При перевірці зондом отримана сигнатура співпадає з еталонною, отже обираємо позитивну гілку, що виходить з лінії 9.

Далі за допомогою зонду проводиться перевірка лінії 7. Сигнатура не співпадає з еталонною. Це може означати, що несправність є на лінії 7 або на лінії 1. Отже треба провести перевірку зондом і на лінії 1. При перевірці сигнатура співпала з еталонною. Тобто несправність заходиться на лінії 7. Перевірка завершена.

3.5 Висновок до розділу 3

У третьому розділі магістерської роботи розглянуто структурні алгоритми пошуку дефектів такі, як методи стиску двійкових послідовностей, зондовий алгоритм пошуку дефектів, зворотнє простеження та умовний зондовий алгоритм пошуку дефектів. Визначені такі поняття,

як тест, сигнатура, об'єкт діагностування, зондова елементарна перевірка, алгоритм діагностування, контрольна точка, наступники та діагностичний експеримент.

Зазначено, що сигнатурний аналіз дозволяє з високою точністю локалізувати несправність і не вимагає високої кваліфікації обслуговуючого персоналу, однак його недоліком є непридатність для захисту від нових вірусів.

Алгоритм діагностування за допомогою зонду не диференціюється чітко на процедури контролю та пошуку дефектів та передбачає напівавтоматичний діалоговий режим перевірки за участю інженера-оператора, що є дуже ефективним способом діагностики. Однак недоліком зондової організації перевірок є необхідність забезпечити хороший контакт в місці з'єднання зонду і досліджуваного елемента.

Визначені властивості матриць досяжності та дерев пошуку несправностей.

Запропоновано використання методів стиску двійкових послідовностей, зондового алгоритму пошуку дефектів, зворотного простеження та умовного зондового алгоритму пошуку дефектів. Для кожного методу наведено приклади їх використання на практиці.

РОЗДІЛ 4
ВИКОРИСТАННЯ ДЕДУКТИВНО-ПАРАЛЕЛЬНОГО МЕТОДУ ДЛЯ
ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ У ЦИФРОВИХ СИСТЕМАХ

4.1 Узагальнена модель дедуктивно-паралельного методу аналізу несправностей

На двійковому тестовому векторі за одну ітерацію обробки схеми можливо вирахувати всі дефекти, використовуючи модель дедуктивно-паралельного синхро-аналізу несправностей дискретного об'єкту.

$$L = T \oplus F, \quad (4.1)$$

де $F = (F_{m+1}, F_{m+2}, \dots, F_i, \dots, F_n)$ ($i = \overline{m+1, n}$) є сукупністю функцій справної поведінки приладу; $Y_i = F_i(X_{i1}, \dots, X_{ij}, \dots, X_{in_i})$ – n_i є вхідним i -им елементом схеми, що для виявлення стану виходу Y_i на тестовому векторі T_t реалізує F_i ; X_{ij} – j -й вхід i -го елемента; m – кількість входів; тест $T = (T_1, T_2, \dots, T_t, \dots, T_k)$ – сукупність двійкових векторів на множині вхідних, внутрішніх та вихідних ліній (виходів), що була в процесі справного моделювання до визначена, впорядкована та об'єднана в матрицю

$$T = [T_{ti}] = \begin{bmatrix} T_{11} & T_{12} & \dots & T_{1i} & \dots & T_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ T_{t1} & T_{t2} & \dots & T_{ti} & \dots & T_{tn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ T_{k1} & T_{k2} & \dots & T_{ki} & \dots & T_{kn} \end{bmatrix} \quad (4.2)$$

за допомогою моделювання функції можна визначити невхідну координату матриці; $T_{ti} = Y_i = F_i(X_{i1}, \dots, X_{ij}, \dots, X_{in_i})$ на тестовому векторі T_t ; $L = (L_1, L_2, \dots, L_t, \dots, L_k)$ – множина дедуктивних схем (моделей), що визначаються за допомогою виразу (4.1), де $L_t = (L_{t1}, L_{t2}, \dots, L_{ti}, \dots, L_{tn})$;

$$L_{ti} = T_t \oplus F_i, \quad (4.3)$$

– дедуктивна функція (ДФ) паралельного моделювання несправностей, що має відповідність справному елементу F_i , на тестовому векторі T_t . Вирахувати список вхідних несправностей, котрі транспортуються на вихід елементів F_i , дає можливість елемент F_i [49].

Синхронність моделі (4.1) може бути виявлена за допомогою умови: $\Delta t = (t_{j+1} - t_j) \gg \tau \gg \tau_i$, коли проміжок часу поміж зміною вхідних наборів $(t_{j+1} - t_j)$, що подаються на схему, є набагато більшим на відміну від найбільшої затримки схеми елементу Це дає можливість виключити час як несуттєвий параметр, тому й використовується в технологіях моделювання [49].

Коли функція приладу задана таблицею істинності, використання формули (4.1) у загальному випадку дає змогу для отримати для тестового вектору T_t таблицю транспортування несправностей. Завдяки даній таблиці можливо записати ДФ моделювання дефектів. Наприклад на рисунку 4.1 перший доданок є тестовим вектором, другий та третій – таблиці істинності та транспортування дефектів.

X1	X2	Y1
0	1	0

 \oplus

X1	X2	Y1
0	0	0
0	1	0
1	0	1
1	1	1

 $=$

X1	X2	Y1
0	1	0
0	0	0
1	1	1
1	0	1

$$L_1 = X_1 X_2 \cup X_1 \bar{X}_2$$

X1	X2	Y1
1	1	1

 \oplus

X1	X2	Y1
0	0	0
0	1	0
1	0	1
1	1	1

 $=$

X1	X2	Y1
1	1	1
1	0	1
0	1	1
0	0	0

$$L_2 = X_1 X_2 \cup X_1 \bar{X}_2 \cup \bar{X}_1 X_2$$

Рисунок 4.1 – ДФ моделювання дефектів

Дедуктивні функції L_1 , L_2 записані по конституантам одиниці таблиць транспортування дефектів у вигляді диз'юнктивної нормальної форми (ДНФ).

Враховуючи розділення тесту на складові вектори рівняння (4.1) отримання ДФ для $T_t \in T$ приймає наступний вид: $L_t = T_t \oplus F$. В разі представлення функціонального опису за допомогою примітивів, що формують стан усіх ліній схеми, наступний вираз виступає у якості

формули перетворення на тестовому векторі T_i справної моделі примітива F_i в дедуктивну функцію L_{ti} :

$$L_{ti} = T_t \oplus F_i = f_{ti}[(X_{i1} \oplus T_{t1}), (X_{i2} \oplus T_{t2}), \dots, (X_{ij} \oplus T_{tj}), \dots, (X_{in_i} \oplus T_{tn_i})] \oplus T_{ti} \quad (4.4)$$

Дана формула є основою дедуктивного аналізу цифрових проектів [50, 51].

На основі виразу (4.4) треба ввести деякі визначення, що необхідні для опису алгоритму аналізу дефектів.

Вектором нульових дефектів $S^0 = (S_1^0, \dots, S_i^0, \dots, S_n^0)$ ($S^1 = (S_1^1, \dots, S_i^1, \dots, S_n^1)$), що перевіряються на тестовому векторі $T_t \in T$, є впорядкована у відповідності до нумерації ліній схеми множина одиничних константних несправностей, де одиничне значення координати вектору $S_i^0 = 1$; ($S_i^1 = 1$) свідчить про те, що перевірка дефекту $\equiv 0 (\equiv 1)$ лінії з номером i на поточному вхідному наборі $T_t \in T$, в іншому випадку при умові, що $S_i^0 = 0$; ($S_i^1 = 0$) – неперевірка цієї несправності.

Вектором нульових дефектів $D^0 = (D_1^0, \dots, D_i^0, \dots, D_n^0)$ ($D^1 = (D_1^1, \dots, D_i^1, \dots, D_n^1)$), що перевірені на тесті T , є впорядкована у відповідності до нумерації ліній схеми множина одиничних константних несправностей, де одиничне значення координати вектору $D_i^0 = 1$; ($D_i^1 = 1$) свідчить про те, що перевірка дефекту $\equiv 0 (\equiv 1)$ лінії з номером i на хоча б на одному наборі $T_t \in T$, в іншому випадку при умові, що $D_i^0 = 0$; ($D_i^1 = 0$) – неперевірка цієї несправності.

Матриця дефектів, що перевіряється на тестовому векторі $T_t \in T$, дефектів $M = [M_{ij}]$, розміром $n \times n$, є форма задання списку одиночних константних несправностей, де першочергово її координата ініціюється відповідно до виразу (4.5).

$$[M_{ij}]|_{(i,j=\overline{1,n})} = \begin{cases} 0 \leftarrow (i \neq j); \\ 1 \leftarrow (i = j). \end{cases} \quad (4.5)$$

В процесі моделювання тестового вектору нульові координати матриці можуть доповнюватись одиницями $M_{ij} = 1$, що відповідає перевірці несправностей, що інверсні станам координат тестового вектору $T_{ti} \in T_t$.

Лінія є невхідною \bar{X} , якщо вона є вихідною Y або внутрішньою Z , тобто не відноситься до зовнішніх входів X схеми. Невхідна лінія є вихідною Y , якщо вона з'єднана с вихідним контактом схеми.

Беручи до уваги вирази, зазначені вище, практична реалізація виразу (4.4) зводиться до наступного алгоритму:

1. Визначення для тестового вектора $t = 0$, що обробляється, початкового значення індексу та ініціалізація векторів перевірених дефектів:

$$\forall i (D_i^0 = 0; D_i^1 = 0) \quad (4.6)$$

2. Визначення номеру наступного вхідного набору $t = t+1$ для $T_t \in T$. Якщо $t > k$ (немає вхідних наборів), то кінець моделювання.

3. Справне моделювання всіх примітивів $F_i (i = \overline{1, n})$ схеми на вхідному наборі $T_t^X \in T_t$ з метою довизначення невхідних координат вектору $T_t^{\bar{X}} \in T_t$:

$$T_t^{\bar{X}} = f(T_t^X, F) \quad (4.7)$$

Умовою переходу до наступного пункту алгоритму є ідентичність вектору справного моделювання ліній в двох сусідніх ітераціях $T_t^r = T_t^{r-1}$.

Аналіз пари сусідніх векторів (T_{t-1}, T_t) використовується для моделювання послідовних схем та організації подій.

Якщо на лініях схеми $[T_{t-1}^X(F_i) \neq T_t^X(F_i)]$ виконується умова – наявність змін на входах елемента, що розглядається – то примітив $F_i (i = \overline{1, n})$ моделюється.

4. У відповідності до виразу (4.6) ініціалізація матриці дефектів, що перевіряються на тест-векторі $M = [M_{ij}]$. Ініціалізація вектору дефектів, що перевіряються, $\forall i (S_i^0 = 0; S_i^1 = 0)$. На основі використання формули (4.4) реконфігурування всіх примітивів $F_i (i = \overline{1, n})$ з метою отримання дедуктивної схеми $L_t \leftarrow \forall i (L_{ti} = T_t \oplus F_i)$.

5. Паралельне моделювання несправностей з метою довизначення координат відповідних невхідним лініям схеми шляхом виконання регістрових операцій над рядками матриці дефектів M , що перевіряється, за допомогою дедуктивних функцій, що були отримані.

6. Формування векторів, що перевіряються несправностей до всіх входів матриці відповідним вихідним лініям схеми за допомогою формули:

$$S^0 = \left(\bigcup_{\forall i \in Y} M_i \right) \cap T_t; S^1 = \left(\bigcup_{\forall i \in Y} M_i \right) \cap \bar{T}_t \quad (4.8)$$

7. Якість тестового вектора $T_t \in T$ при ідентичності списків несправностей у двох сусідніх ітераціях $(S^0 \cup S^1)^{r-1} = (S^0 \cup S^1)^r$ (де r - це індекс ітерації) визначається формулою:

$$Q(T_t) = \frac{1}{2n} [\sum_{i=1}^n (S_i^0 + S_i^1)] \quad (4.9)$$

та здійснюється перехід до наступного пункту. Інакше виконується виключення дефектів із процесу моделювання за правилом:

$$(S_i^0 = S_i^1 = 0) \leftarrow \forall i [(S_i^0 \cup S_i^1 = 1)^{r-1} \& (S_i^0 \cup S_i^1 = 0)^r], \quad (4.10)$$

якщо спостерігається зникнення несправностей, що перевіряються в ітерації r порівняно з $r-1$:

$$\exists i [(S_i^0 \cup S_i^1 = 1)^{r-1} \& (S_i^0 \cup S_i^1 = 0)^r] \quad (4.11)$$

здійснюється перехід до пункту п'ять.

8. Формування векторів перевірених несправностей відповідно до виразу

$$D^0 = D^0 \cup S^0, D^1 = D^1 \cup S^1 \quad (4.12)$$

та визначення якості тесту за формулою:

$$Q(T_t) = \frac{1}{2n} [\sum_{i=1}^n (D_i^0 + D_i^1)] \quad (4.13)$$

Запропонований алгоритм та його реалізація орієнтована як на табличний опис примітивів довільної складності RTL-рівня, так і на вентильне представлення цифрових схем.

Швидкодія алгоритму інваріантна компілятивним та інтерпретативним моделям цифрових пристроїв. Однак з позиції програмування чисто інтерпретативна реалізація є найбільш технологічною.

Блок-схема даного алгоритму представлена на рисунку 4.2.

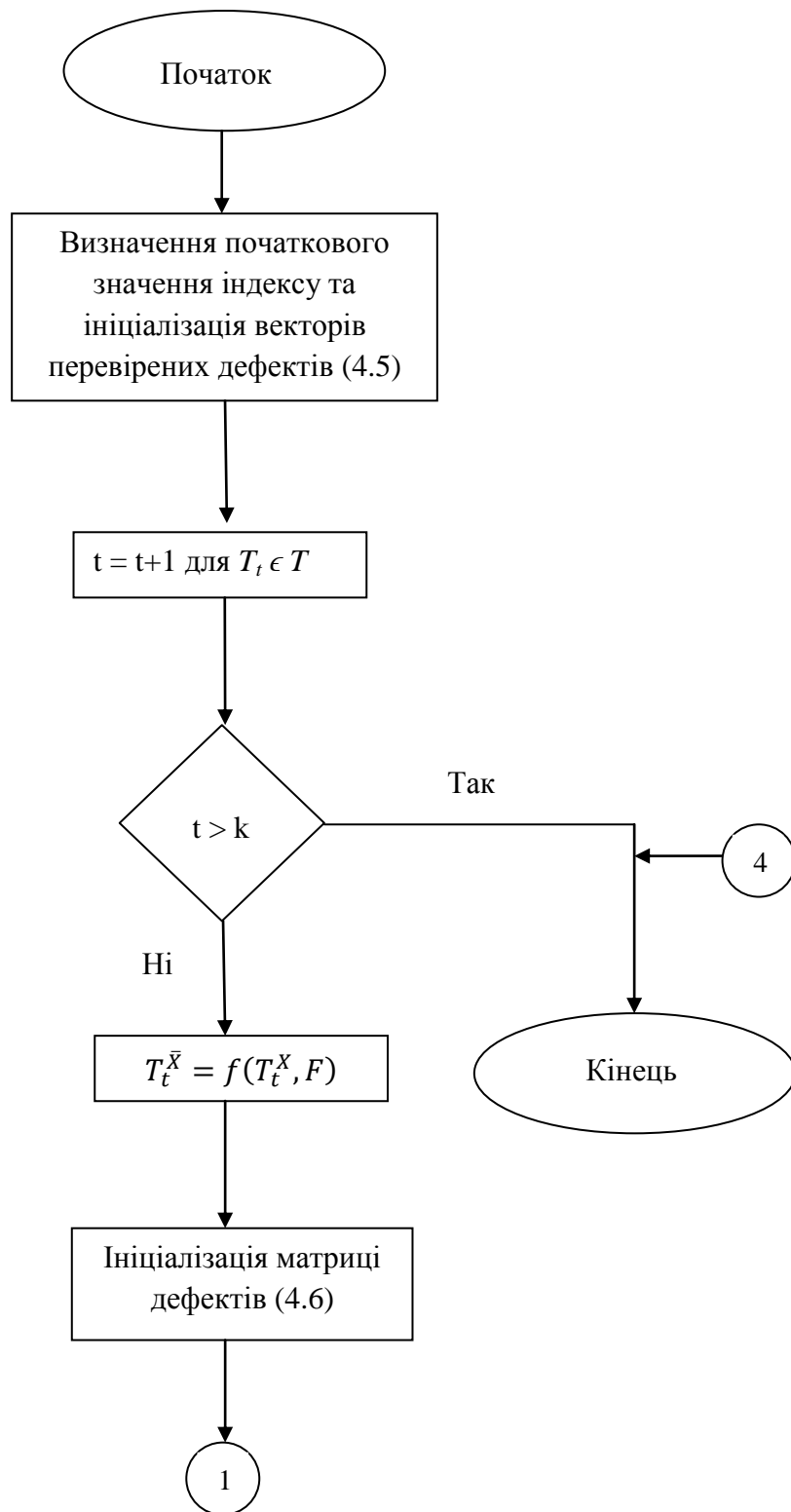


Рисунок 4.2, аркуш 1 – Блок-схема алгоритму реалізації виразу (4.4)

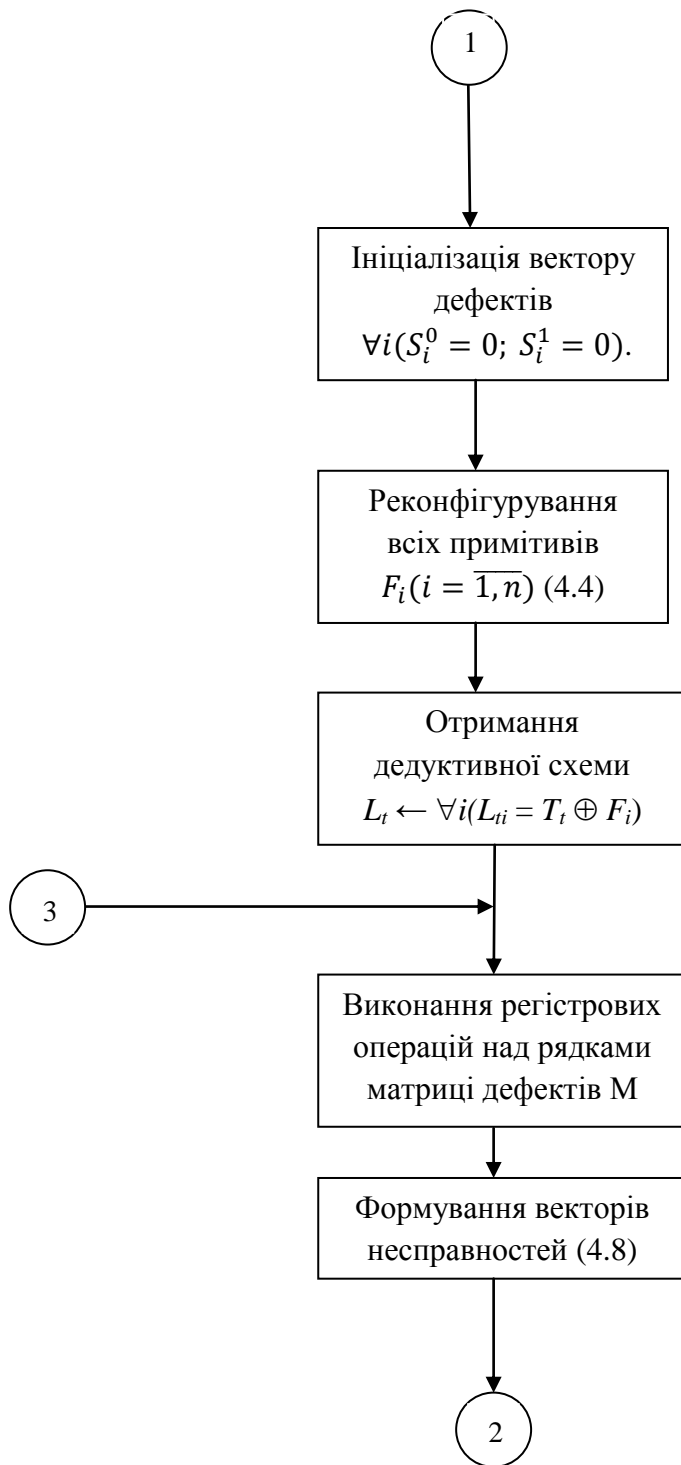


Рисунок 4.2, аркуш 2

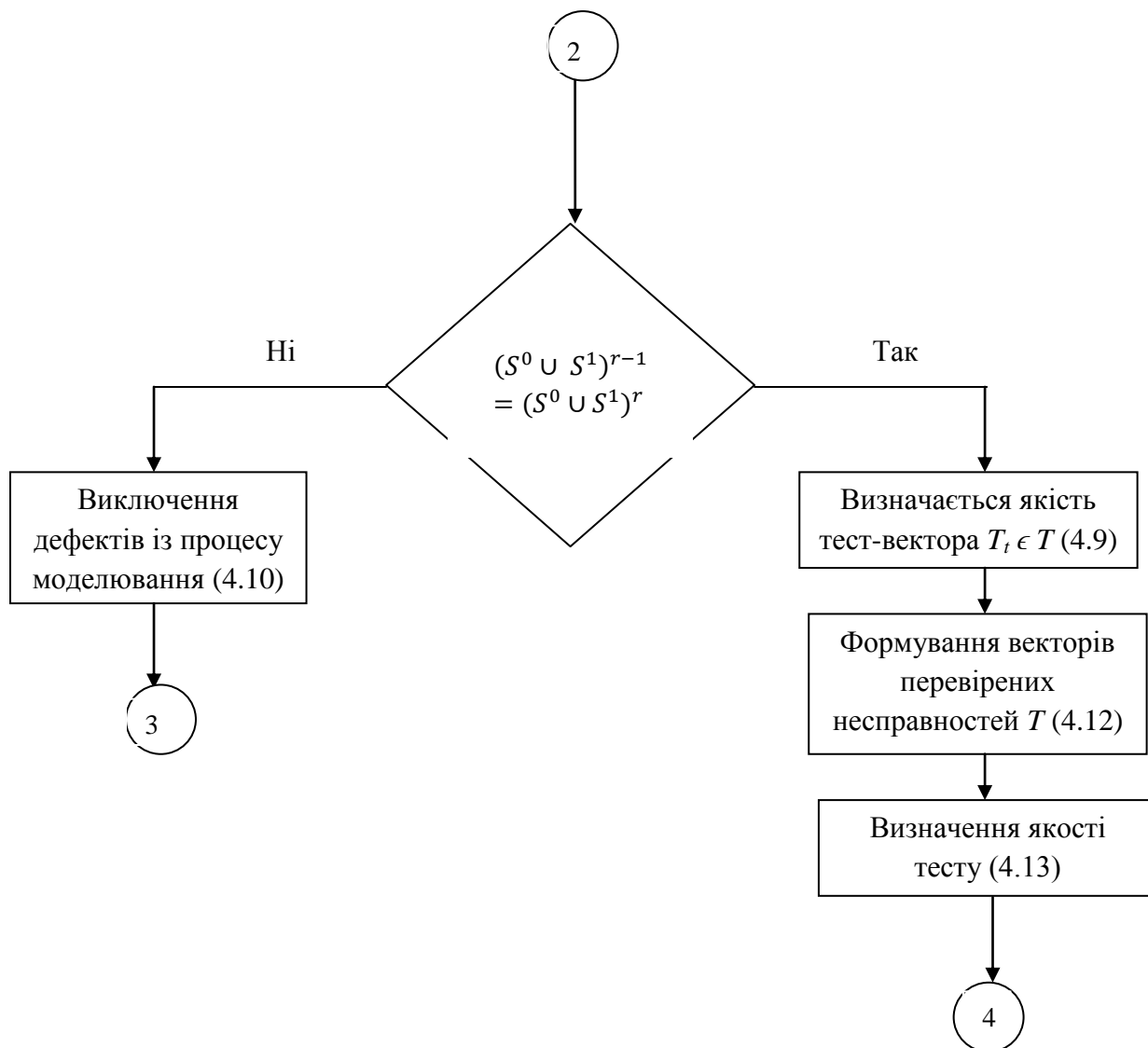


Рисунок 4.2, аркуш 3

Нижче розраховані дедуктивні функції паралельного моделювання несправностей на вичерпному тесті для базису логічних елементів І, АЛЕ та НІ.

Враховуючи вираз (4.4) розрахунок для функції АЛЕ буде виглядати так:

$$\begin{aligned}
 & L[T = (00,01,10,11), F = (X_1 \cup X_2)] = \\
 & = L\{(\overline{x_1x_2} \cup \overline{x_1}x_2 \cup x_1\overline{x_2} \cup x_1x_2) \cap [X_1 \oplus T_{t_1} \cup X_2 \oplus T_{t_2}] \oplus T_{t_3}\} = \\
 & = (\overline{x_1} \overline{x_2})\{[(X_1 \oplus 0) \cup (X_2 \oplus 0)] \oplus 0\} \cup (\overline{x_1}x_2)\{[(X_1 \oplus 0) \cup (X_2 \oplus 1)] \oplus 1\} \cup \\
 & \cup (x_1\overline{x_2})\{[(X_1 \oplus 1) \cup (X_2 \oplus 0)] \oplus 1\} \cup (x_1x_2)\{[(X_1 \oplus 1) \cup (X_2 \oplus 1)] \oplus 1\} = \\
 & = (\overline{x_1} \overline{x_2})(X_1 \cup X_2) \cup (\overline{x_1} x_2)(\overline{X_1} \cap X_2) \cup (x_1 \overline{x_2})(X_1 \cap \overline{X_2}) \cup (x_1x_2)(X_1 \cap X_2).
 \end{aligned} \tag{4.14}$$

Аналогічний розрахунок виконується для функції І:

$$\begin{aligned}
& L[T = (00,01,10,11), F = (X_1 \cap X_2)] = \\
& = L\{(\overline{x_1 x_2} \cup \overline{x_1} x_2 \cup x_1 \overline{x_2} \cup x_1 x_2) \cap [X_1 \oplus T_{t_1} \cap X_2 \oplus T_{t_2}] \oplus T_{t_3}\} = \\
& = (\overline{x_1} \overline{x_2})\{[(X_1 \oplus 0) \cap (X_2 \oplus 0)] \oplus 0\} \cup (\overline{x_1} x_2)\{[(X_1 \oplus 0) \cap (X_2 \oplus 1)] \oplus 0\} \cup \\
& \cup (x_1 \overline{x_2})\{[(X_1 \oplus 1) \cap (X_2 \oplus 0)] \oplus 0\} \cup (x_1 x_2)\{[(X_1 \oplus 1) \cap (X_2 \oplus 1)] \oplus 1\} = \\
& = (\overline{x_1} \overline{x_2})(X_1 \cap X_2) \cup (\overline{x_1} x_2)(X_1 \cap \overline{X_2}) \cup (x_1 \overline{x_2})(\overline{X_1} \cap X_2) \cup (x_1 x_2)(X_1 \cup X_2).
\end{aligned} \tag{4.15}$$

В даному випадку $T_t = T_{t_1}, T_{t_2}, T_{t_3}$, $(t = \overline{1,4})$ є тестовий вектор, що має три координати, де остання з них визначає стан виходу елементу І (АБО). У наступному перетворенні $T_t = T_{t_1}, T_{t_2}$, $(t = \overline{1,2})$ є тестовим вектором, що має дві координати, де друга є станом виходу інвертору:

$$\begin{aligned}
L[T = (0,1), F = \overline{X_1}] &= L\{(\overline{x_1} \cup x_1)[(X_1 \oplus T_{t_1}) \oplus T_{t_2}]\} = \\
&= \overline{x_1}[(X_1 \oplus 0) \oplus 1] \cup x_1[(X_1 \oplus 1) \oplus 0] = \\
&= \overline{x_1} \overline{X_1} \cup x_1 \overline{X_1} = \overline{x_1} X_1 \cup x_1 X_1.
\end{aligned} \tag{4.16}$$

Останній вираз вказує на інваріантність інверсії до вхідного набору для транспортування дефектів. Вона перетворюється в повторювач, тому дана функція є на виходах дедуктивних елементів.

На рисунку 4.3 представлена апаратна реалізація універсального функціонального примітиву (симулятора) дедуктивно-паралельного аналізу несправностей.

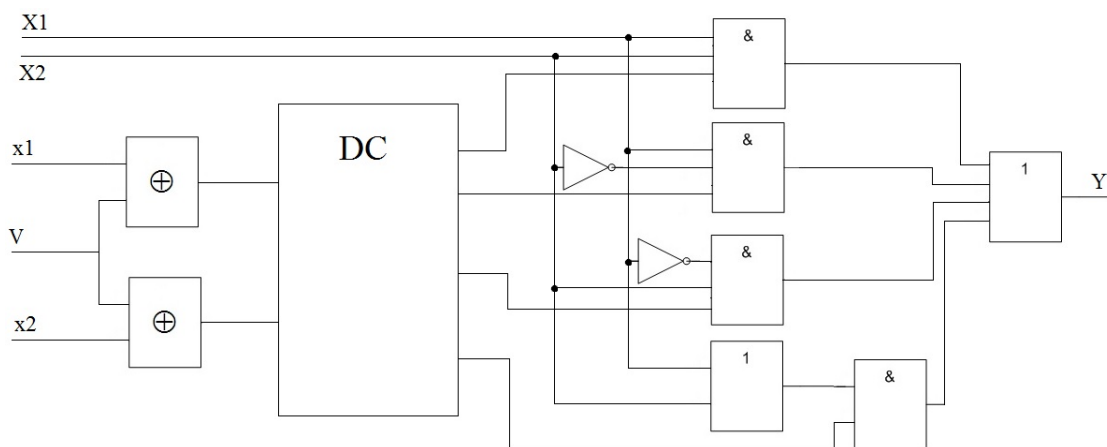


Рисунок 4.3 – Симулятор несправних примітивів

У симуляторі представлені регістри (X_1, X_2) та булеві (x_1, x_2) входи, логічні функції (елементи) (І, АБО) вихідна регістрова змінна Y .

Таблиця 4.1 ілюструє паралельне моделювання вхідних 4-розрядних векторів несправностей для отримання на виході Y множини дефектів для логічних елементів, що перевіряються.

Таблиця 4.1 – Моделювання вхідних 4-розрядних векторів несправностей

I/АБО	x1x2	X1	X2	Y
0	00	0111	1011	0011
1	00	0111	1011	1111
0	11	0101	0110	0111
1	11	1101	0111	0101

Даний симулятор дає можливість трансформувати вентильну модель F справної поведінки схеми у дедуктивну модель L , що не передбачає в процесі моделювання використання вентильної моделі. Як апаратна модель ДФ симулятор орієнтований на створення вбудованих засобів дедуктивно-паралельного моделювання, що в порівнянні з програмною реалізацією підвищує швидкодію аналізу в 10-1000 разів.

Мати внутрішню модель справної поведінки схеми необхідно згідно з третім пунктом алгоритму дедуктивно-паралельного аналізу у вигляді відповідних структур даних, розміщених в оперативній пам'яті та орієнтовані на обробку елемента цифрового виробу (рисунок 4.4).

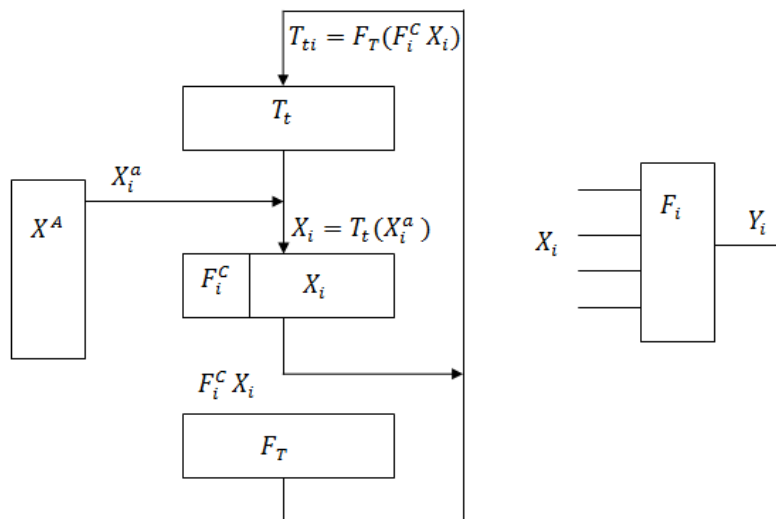


Рисунок 4.4 – Структурна модель аналізу елемента

Насамперед процедура виявлення стану координати тестового вектору $T_{ti} \in T$ визначається формулою (4.17):

$$T_{ti} = F_i(X_{i1}, \dots, X_{ij}, \dots, X_{in_i}), \quad (4.17)$$

де Y_i – вихід логічного елемента;

$F_i (F_i^C)$ – двійковий код елемента або його ідентифікатор за його вхідними значеннями, що представлені вектором $X_i = (X_{i1}, \dots, X_{ij}, \dots, X_{in_i})$ за умовою, що F_T – таблиця істинності сукупності булевих функцій для опису елементів схеми, вхідний набір якої формується за допомогою $(F_i^C * X_i)$.

З чого виходить функція (4.18).

$$T_{ti} = F_T(F_i^C X_i), \quad (4.18)$$

Необхідно сформувати двійковий вектор значень вхідних змінних $X_i = T_t(X_i^a)$; $X_{ij} = T_t(X_{ij}^a)$ елемента F_i для того, щоби виявити стан координати $T_{ti} \in T$. Для цього використовуються їх адреси $X_i^a = (X_{i1}^a, \dots, X_{ij}^a, \dots, X_{in_i}^a)$ для витягу станів з вектору T_t . Наступним кроком буде конкатенація отриманого вектору X_i з двійковим кодом типу функції F_i^C з метою отримання вхідного слова $(F_i^C * X_i)$ для узагальненої таблиці істинності F_T . В узагальненій таблиці істинності у стовпці Y_i знаходиться стан координати $T_{ti} \in T$.

На рисунку 4.5 представлена модель аналізу дедуктивно-паралельним методом, де містяться, окрім структури, яка бере участь у справному моделюванні, два додаткових модулі М та L.

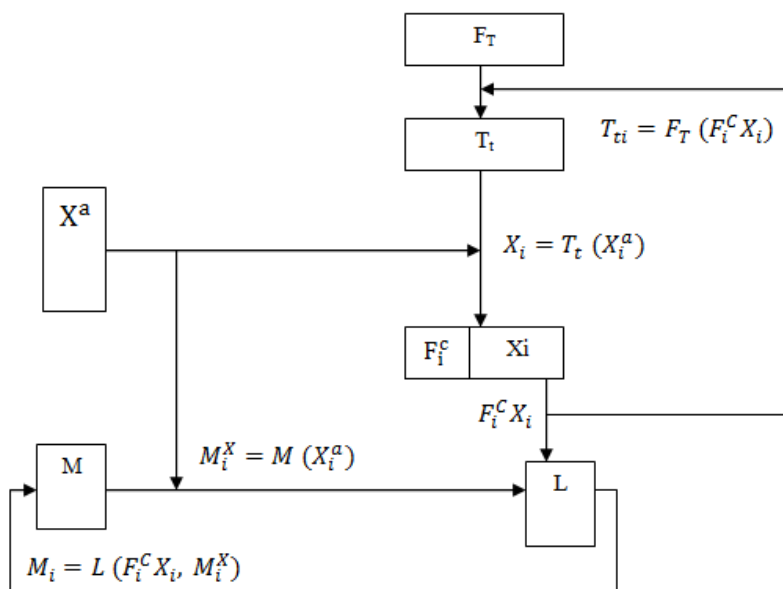


Рисунок 4.5 – Модель дедуктивно-паралельного аналізу

де M і L – два додаткових модулі;

Y_i – вихід логічного елементу;

F_i^C – двійковий код елементу або його ідентифікатор;

F_T – таблиця істинності сукупності булевих функцій для опису елементів схеми;

$(F_i^C * X_i)$ – вхідний набір;

$$M_i = L(F_i^C X_i, M_i^X) \quad (4.19)$$

Отриманий з F аналітичний вираз (4.19) для визначення перевіряємих векторів несправностей, об'єднаних в матрицю M за допомогою набору дедуктивних функцій L .

В функції моделюється підмножина векторів несправностей $M_i^X = M(X_i^a) = \{M(X_{i1}^a), \dots, M(X_{ij}^a), \dots, M(X_{in_i}^a)\}$, що відповідають адресам вхідних змінних $X_i^a \in X^a$ i -го елементу на дедуктивній функції $L(F_i^C X_i) \in L$, що ідентифікується адресою $(F_i^C X_i)$, що конкатенується та реалізуються за допомогою компіляції для паралельного виконання регістрових операцій над векторами з множини M_i^X .

Таким чином виходить, що необхідно знайти адресу дедуктивної функції L_i для визначення стану вхідного вектору несправностей M_i . Для цього треба використати отриману для справного моделювання конкатенацію двійкових слів $F_i^C X_i$. Вхідні змінні для елементу L_i є регістровими. В теорії їх розмірність дорівнює кількості ліній в цифровому виробі. В подальшому виконується послідовне виконання $(n-1)$ регістрових операцій над вхідними векторами M_i^X . Результат записується в матрицю M у вигляді рядка M_i . Вхідна змінна X_{ij} може бути інверсною. Тоді перед виконанням двійкової операції необхідно виконати інверсію вмісту регістрової змінної:

$$\overline{M(X_{ij})} = \overline{M}(X_{ij}^a). \quad (4.20)$$

Як приклад розглянуто узагальнену таблицю істинності (таблиця 4.2) для визначення справної поведінки і вибору бінарної адреси дедуктивного функціонального елементу аналізу несправностей.

В таблиці 4.2 F_i – код справної функції; I – умовне позначення; $(x1x2)$ – вхідні двійкові комбінації таблиці істинності для кожної з чотирьох функцій; Y_i – стовпець стану виходу справної функції; L_{ti} – код адреси однієї з чотирьох компілятивних дедуктивних функцій.

Таблиця 4.2 – Узагальнена таблиця істинності

I	F _i	x1x2	Y _i	L _{ti}	I	F _i	x1x2	Y _i	L _{ti}
∩	00	00	0	00	∩̄	10	00	1	00
	00	01	0	01		10	01	1	01
	00	10	0	10		10	10	1	10
	00	11	1	11		10	11	0	11
∪	01	00	0	11	∪̄	11	00	1	11
	01	01	1	10		11	01	0	10
	01	10	1	01		11	10	0	01
	01	11	1	00		11	11	0	00

Відповідно до L_{ti} , що визначені відповідно до функції (4.3), маємо:

$$L = \begin{cases} 00 \rightarrow X_1 \cap X_2; \\ 01 \rightarrow X_1 \cap \bar{X}_2; \\ 10 \rightarrow \bar{X}_1 \cap X_2; \\ 11 \rightarrow X_1 \cap X_2. \end{cases} \quad (4.21)$$

За допомогою виразу (4.22) вираховується обчислювальна складність обробки цифрової схеми, що складається з n -ої кількості двовходових вентилів.

$$Q = \left[(2K + A) + A + \frac{2n\tau}{W} \right] = \left[2(K + A) + \frac{2n\tau}{W} \right] \times n, \quad (4.22)$$

де K – час конкатенації бітів для отримання адреси стану виходу примітива;

A – час, за який буде обрано біт (вміст комірки за її адресою);

τ – час, за який виконується регістрова операція (І, АЛЕ, НІ);

W – розрядність регістру.

Якщо брати до уваги, що перший доданок в порівнянні з другим є нескінченно малим, то обчислювальна складність обробки цифрового виробу пропорційна квадрату кількості вентилів:

$$Q = \frac{2n^2\tau}{W}. \quad (4.23)$$

4.2 Дедуктивний і дедуктивно-паралельний методи структурного аналізу схеми та моделювання графових структур

Припустимо є орієнтований граф G , в якому є деяка вершина $V_j \in V$. Відповідно до суміжних вершин-попередників (прообрази $f^{-1}(V_j)$) вище зазначена вершина є образом. У вершинах-попередниках $f^{-1}(\dots)$ є оператором взяття прообразу (рисунок 4.6).

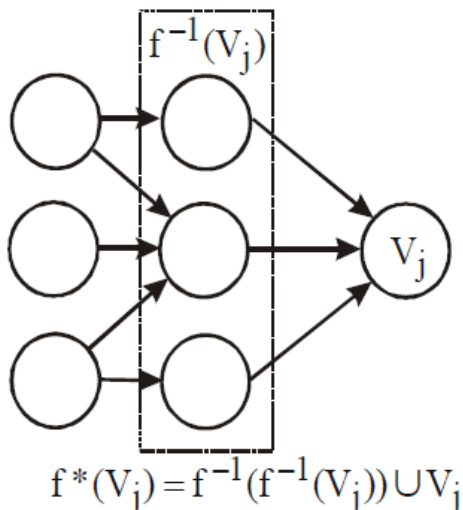


Рисунок 4.6 – Суміжність

Для демонстрування використання дедуктивного методи в структурному аналізі необхідно виконати перетворення графу до схеми-моделі, що складається з логічних елементів АБО (АБО-структура). Це виконується за правилами:

- 1) Зовнішніми входами схеми стають вершини, що містять у собі виключно вхідні дуги;
- 2) решта вершин перетворюються на виходи, іншими словами – елементами АБО;
- 3) функціональні зв'язки суміжних вершин формуються з дуг графу.

Таким чином, граф трансформується в АБО-структуру, де вершини графу стають лініями схеми. У схемі кількість входів дорівнює числу суміжних вершин-попередників.

На рисунку 4.7 зображений приклад дев'яти-вершинного графу, що трансформується в логічну схему з трьома входами та шістьма елементами АБО.

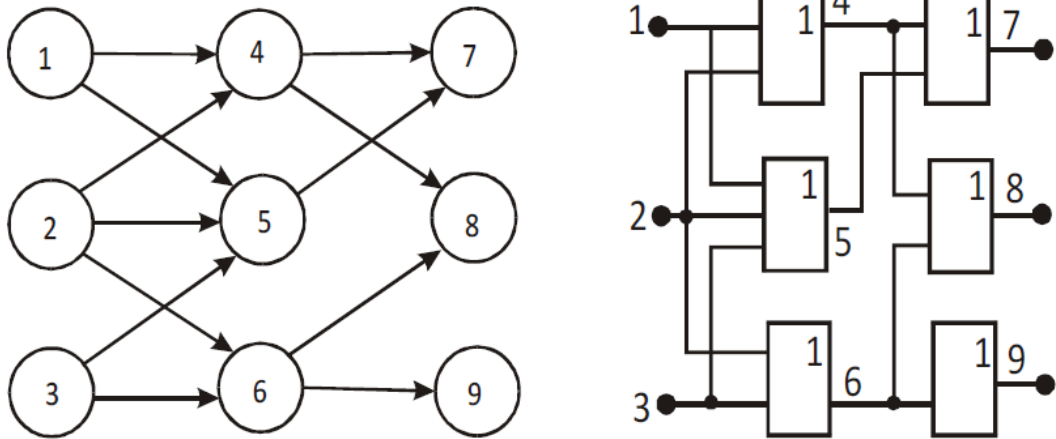


Рисунок 4.7 – Трансформація графу в логічну схему

Процедура аналізу графу (при умові, що він орієнтований та не має глобальних контурів зворотного зв'язку) за допомогою дедуктивного методу полягає в тому, що необхідно на основі дедуктивної обробки $V_j \in V$ провести одноразовий прохід всіх його вершин. Цей процес складається з трьох кроків, описаних формулами (4.24).

$$\begin{aligned}
 1) V^j &= \bigcup_{C_{n_j}^2} [f^*(f_p^{-1}(V_j)) \bigcap_{i=1, n_j-1}^{p-i, q=i+1, n_j} f^*(f_q^{-1}(V_j))]; \\
 2) V^j &= V^j \setminus \bigcup_{C_{m_j}^2} [f^*(V_p^j) \bigcap_{i=1, m_j-1}^{p-i, q=i+1, m_j} f^*(V_q^j)]; \\
 3) V_j &= [\bigcup_{i=1}^{n_j} f^*(f_i^{-1}(V_j)) \cup V_j] \setminus V^j,
 \end{aligned} \tag{4.24}$$

де $f^*(f^{-1}(V_j))$ є екстраобразами прообразів $f^{-1}(V_j)$, кількість котрих для вершини V_j дорівнює n_j .

Призначення першого рівняння полягає у визначенні збіжних розгалужень за допомоги виконання операції об'єднання перетинів кожної пари екстраобразів для прообразів вершини, що аналізується. Кількість пар екстраобразів дорівнює кількості комбінацій – $C_{n_j}^2$.

Друге рівняння призначено для виключення вершин, що не є вершинами згідно з формулою (4.2), зі списку збіжних розгалужень (ЗР), де $f * (V_i^j)$ є екстраобразом вершини, що входить до переліку кандидатів в ЗР при аналізі $V_j \in V$. Кількість даних вершин дорівнює m_j .

Призначення третього рівняння полягає в формуванні екстраобразу вершини $V_j \in V$, що обробляється.

Результати (4.25) послідовної обробки всіх вершин графу (рисунок 4.7) за правилами, наведеними у формулі (4.24) наведені нижче:

$$\begin{aligned}
 V^1 &= \emptyset; V_1 = \{1\}; V^2 = \emptyset; V_2 = \{2\}; V^3 = \emptyset; V_3 = \{3\}; \\
 V^4 &= (V_1 \cap V_2) = \emptyset; V_4 = \{1,2,4\}; \\
 V^5 &= (V_1 \cap V_2) \cup (V_1 \cap V_3) \cup (V_2 \cap V_3) = \emptyset; V_5 = \{1,2,3,5\}; \\
 V^6 &= (V_2 \cap V_3) = \emptyset; V_6 = \{1,2,3,5\}; \\
 V^7 &= (V_4 \cap V_5) = \{1,2\}; V_7 = \{1,2,3,4,5,7\} \setminus \{1,2\} = \{3,4,5,7\}; \\
 V^8 &= (V_4 \cap V_6) = \{2\}; V_8 = \{1,2,3,4,6,8\} \setminus \{2\} = \{1,3,4,6,8\}; \\
 V^9 &= \emptyset; V_9 = \{2,3,6,9\};
 \end{aligned} \tag{4.25}$$

Об'єднання всіх підмножин ЗР графа цифрової структури у загальному вигляді представлено формулою (4.26), а для прикладу з графом на рисунку 4.7 показано у виразі (4.27).

$$V^{RC} = \bigcup_{j=1}^n V^j \tag{4.26}$$

$$\begin{aligned}
 V^{RC} &= V^1 \cup V^2 \cup V^3 \cup V^4 \cup V^5 \cup V^6 \cup V^7 \cup V^8 \cup V^9 = \\
 &= \emptyset \cup \emptyset \cup \emptyset \cup \emptyset \cup \emptyset \cup \emptyset \cup \{1,2\} \cup \{2\} \cup \emptyset = \{1,2\}.
 \end{aligned} \tag{4.27}$$

Беручи до уваги, що вже зазначені як ЗР лінії є надлишковими, їх можна виключити з подальшого аналізу структури та з переліку обробки кожної вершини графу. Треба зазначити, що визначені в процесі обробки ЗР необхідно накопичувати до окремої множини V^{RC} , з чого виходить, що формули (4.24) треба модифікувати до наступного (4.28) вигляду. Формули (4.28) використовуються з метою зменшення потужності прообразів вершини, що обробляється. Чим вище коефіцієнт розгалуженості ліній попередників у переліку V^{RC} , тим суттєвіша потужність.

$$\begin{aligned}
1) V^j &= \bigcup_{C_{n_j}^2} [f^*(f_p^{-1}(V_j)) \bigcap_{\substack{p=i \\ q=i+1, n_j}}^{i=1, n_j-1} f^*(f_q^{-1}(V_j))]; \\
2) V^j &= V^j \setminus \bigcup_{C_{m_j}^2} [f^*(V_p^j) \bigcap_{\substack{p=i \\ q=i+1, m_j}}^{i=1, m_j-1} f^*(V_q^j)]; \\
3) V^{RC} &= V^{RC} \cup V^j; \\
4) V_j &= [\bigcup_{i=1}^{n_j} f^*(f_i^{-1}(V_j)) \cup V_j] \setminus V^{RC}.
\end{aligned} \tag{4.28}$$

Окремо розглядаючи паралельно-дедуктивний метод моделювання графів цифрових систем, зазначимо, що вектор $V_j \in V$ виступає в якості ідентифікатору V_j , та в j -тій позиції дорівнює 1. Тоді одиничною матрицею (4.29) можна виразити сукупність векторів, що є усіма множинами вершин графу.

$$V = \|V_{ij}\|_{(i,j=\overline{1,n})}; V_{ij} = \begin{cases} 1 \leftarrow i = j; \\ 0 \leftarrow i \neq j. \end{cases} \tag{4.29}$$

З цього виконується модифікація формул (4.24) в формули (4.30) та аналогічно – формул (4.28) в формули (4.31), де над відповідними векторами матриці V виконуються операції диз'юнкції та кон'юнкції.

$$\begin{aligned}
1) V^j &= \bigvee_{C_{n_j}^2} [f^*(f_p^{-1}(V_j)) \bigwedge_{\substack{p=i \\ q=i+1, n_j}}^{i=1, n_j-1} f^*(f_q^{-1}(V_j))]; \\
2) V^j &= V^j \wedge \neg \{ \bigvee_{C_{m_j}^2} [f^*(V_p^j) \bigwedge_{\substack{p=i \\ q=i+1, m_j}}^{i=1, m_j-1} f^*(V_q^j)] \}; \\
3) V_j &= [\bigvee_{i=1}^{n_j} f^*(f_i^{-1}(V_j)) \vee V_j] \wedge \bar{V}^j,
\end{aligned} \tag{4.30}$$

Вектор ЗР V^{RC} ініціалізується нулями та зображений у таблиці 4.4.

Таблиця 4.4 – Вектор ЗР V^{RC}

$$V^{RC} = \begin{array}{c|cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

2. Матриця V^1 сформована відповідно до структури графу після обробки вершин і зображена у таблиці 4.5.

Таблиця 4.5 – Матриця V^1

V^1	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	1	1	0	1	0	0	0	0	0
5	1	1	1	0	1	0	0	0	0
6	0	1	1	0	0	1	0	0	0
7	0	0	1	1	1	0	1	0	0
8	1	0	1	1	0	1	0	1	0
9	0	1	1	0	0	1	0	0	1

3. Вектор V^{RC} є остаточним результатом після процесу обробки вершин графу та накопичення ЗР. Він зображений у таблиці 4.6, де 1 означає належність лінії до підмножини ЗР.

Таблиця 4.6 – Остаточний результат, вектор V^{RC}

$$V^{RC} = \begin{array}{c|cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

4.3 Дедуктивний аналіз та синтез дедуктивних компонентів для SoC

Дедуктивно-паралельний аналіз дефектів орієнтований на обробку великих проектів вентильного та реєстрового рівней опису з метою отримання таблиці несправностей та оцінки якості покриття тестом дефектів заданого класу. Об'єкт тестування представлений у формі структур, таблиць, булевих рівнянь, кубічних покриттів, що реалізує складну цифрову систему,

вбудовану в кристали SoC. Дедуктивний метод моделювання несправностей об'єднує в собі переваги раніше розглянутих в цьому розділі методів: дедуктивного виявлення переліків несправностей, що є ефективним с точки зору математики, та виконання паралельних процедур, що орієнтовані на обробку з високою швидкістю ЦП вентиляного, реєстрового та системного рівнів опису.

З точки зору теорії множин ідеєю дедуктивного методу моделювання несправностей є розрахунок вихідної підмножини дефектів, що перевіряються за одну ітерацію аналізу примітиву. Аналіз можливих дефектів будь-якого пристрою за допомогою найбільшої дедуктивної моделі аналізу здійснюється за допомоги формули (4.32).

$$\begin{aligned}
 L_r &= \left\{ \bigcup_{Z^\emptyset} \left[\left(\bigcap_{X^\emptyset} L_j \right) \setminus \left(\bigcup_{X^\emptyset} L_j \right) \right] \right\} \cup \{ \equiv \bar{T}_r^Z \}; \\
 X^\emptyset &= \forall_j (T_j \cap C_{ij}^X = \emptyset); \\
 X^{\bar{\emptyset}} &= \forall_j [(T_j \cap C_{ij}^X \neq \emptyset) \& (C_{ij}^X \neq X)]; \\
 Z^\emptyset &= \forall_i (T_r \cap C_{ir}^Z = \emptyset); \\
 i &= \overline{1, n}; j = \overline{1, m}; r = \overline{1, k},
 \end{aligned} \tag{4.32}$$

де L_r – перелік несправностей;

n – кількість рядків (кубів);

m – кількість вхідних ліній;

k – кількість вихідних ліній у примітиві.

Дедукція у загальному значенні є висновком від загального до приватного у системі математичних доказів. Дедукція з точки зору аналізу несправностей цифрових систем є знаходженням закономірностей в галузі алгебри-логіки, що дозволяє використовувати отримані раніше складні аналітичні моделі, що багаторазово використовуються з метою моделювання несправностей для обробки цифрових систем. Першочергово кожен дефект необхідно описати таблицею істинності, булевою функцією та графом переходів. Дана модель аналізу несправностей дозволяє за одну ітерацію обробки цифрової схеми вирахувати всі можливі несправності, що перевіряються на тестовому векторі. Дедуктивний аналіз дефектів цифрових систем має математичну модель $T \oplus C = L$, що можна виразити матричним рівнянням (4.33).

$$\begin{aligned}
 1) L = f(x, y, a, b) &= (\bar{x}\bar{y} \cap ab) \cup (y \cap a\bar{b}) \cup (x \cap \bar{a}b) \cup (xy \cap b) \cup \\
 &\cup (xy \cap a) = (\bar{x}\bar{y} \cap ab) \cup (x \cap \bar{a}b) \cup (y \cap a\bar{b}) \cup [(xy \cap (a \cup b))] \\
 2) L = f(x, y, a, b) &= (\bar{x}\bar{y} \cap ab) \cup (y \cap a\bar{b}) \cup (x \cap \bar{a}b) \cup (xy \cap a) = \\
 &= (\bar{x}\bar{y} \cap ab) \cup (x \cap \bar{a}b) \cup [(ya \cap (x \cup \bar{b}))] \\
 3) L = f(x, y, a, b) &= (\bar{x}\bar{y} \cap ab) \cup (y \cap a\bar{b}) \cup (x \cap \bar{a}b) \cup (xy \cap b) \\
 &\cup (xy \cap a) = (\bar{x}\bar{y} \cap ab) \cup [(ya \cap (x \cup \bar{b}))] \cup [(xb \cap (y \cup \bar{a}))].
 \end{aligned}
 \tag{4.34}$$

Результати мінімальних по Квайну функцій апаратно реалізуються у наступні схеми: на рисунку 4.8 представлена схема дедуктивного примітиву елементу АБО, на рисунку 4.9 – елементу виключне АБО (складання за модулем 2 – XOR), а на рисунку 4.10 – елементу І.

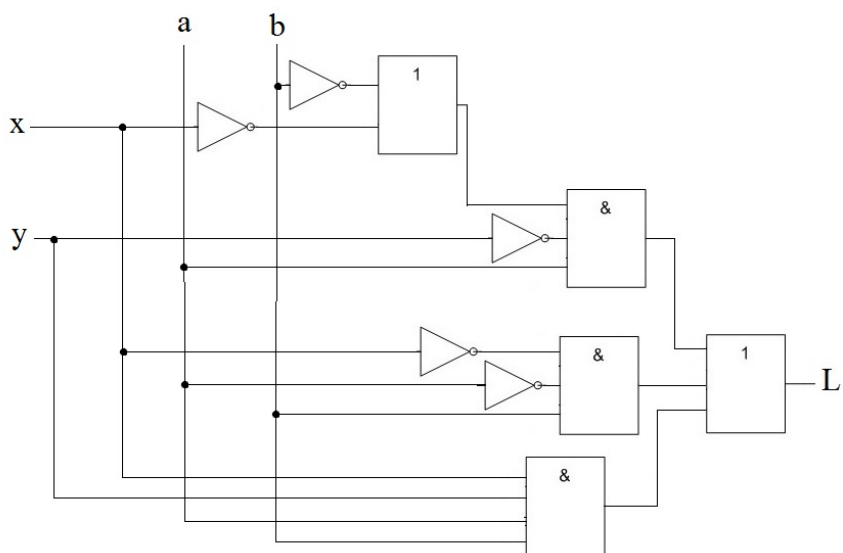


Рисунок 4.8 – Схема дедуктивного примітиву елементу АБО

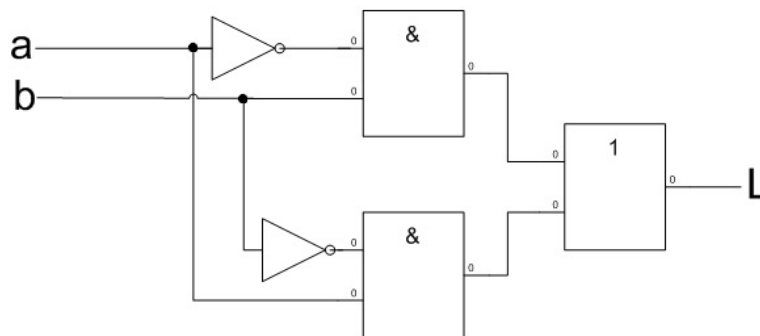


Рисунок 4.9 – Схема дедуктивного примітиву елементу XOR

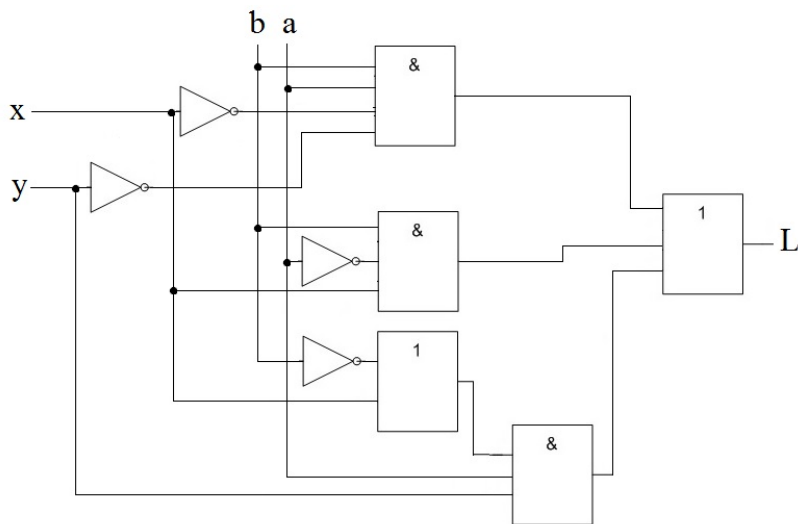


Рисунок 4.10 – Схема дедуктивного примітиву елементу I

Регістровий рівень опису компонентів цифрової системи відрізняється функціональною складністю, що впливає на розмір таблиць істинності та кубічних покриттів. На цьому рівні розглядаються такі функціональності, як тригери, засувки, лічильники, регістри, мультиплексори, шинні структури. Наприклад, синтез дедуктивної функції D-тригера з трьома булевими та трьома регістровими входніми лініями $Q = DC \cup \overline{C\overline{D}Q}(t-1)$, функція якого представлена картою Карно в таблиці 4.8.

Таблиця 4.8 – Карта Карно D-тригера

(T)\(X)	000	001	011	010	110	111	101	100
000	0	1	1	0	1	1	0	0
001	0	1	1	0	0	0	1	1
011	0	1	1	0	1	1	0	0
010	0	1	1	0	0	0	1	1
110	0	0	1	1	1	0	0	1
111	0	0	1	1	0	1	1	0
101	0	0	1	1	1	0	0	1
100	0	0	1	1	0	1	1	0

$$\begin{aligned}
 L = f(c, d, q, C, D, Q) = & (\bar{c} \cap \overline{C\overline{D}Q}) \cup (\overline{C\overline{D}Q}) \cup (c \cap \overline{C\overline{D}Q}) \cup \\
 & \cup (\bar{c}\bar{d}\bar{q} \cap CD) \cup (\bar{c}\bar{d}\bar{q} \cap C\overline{D}) \cup (\bar{c}d\bar{q} \cap CD) \cup (\bar{c}d\bar{q} \cap C\overline{D}) \\
 & \cup (cd\bar{q} \cap C\overline{Q}) \cup (cd\bar{q} \cap CQ) \cup (c\bar{d}\bar{q} \cap C\overline{Q}) \cup (c\bar{d}\bar{q} \cap CQ)
 \end{aligned}
 \tag{4.35}$$

4.4 Висновки до розділу 4

У четвертому розділі магістерської роботи розглянутий паралельно-дедуктивний метод виявлення дефектів у цифрових системах. Розглянута узагальнена модель дедуктивно-паралельного аналізу несправностей, у якій був запропонований алгоритм, орієнтований як на табличний опис примітивів довільної складності RTL-рівня, так і на вентильне представлення цифрових схем. Швидкодія алгоритму інваріантна компілятивним та інтерпретативним моделям цифрових пристроїв. Однак з позиції програмування чисто інтерпретативна реалізація є найбільш технологічною.

Запропонований дедуктивний метод структурного аналізу, дедуктивно-паралельний метод моделювання графових структур. Головними перевагами модифікованих для дедуктивного методу структурного аналізу формул є те, що операції над вершинами графу виконуються паралельно при програмному препроцесорі моделювання. Це підвищує швидкодію аналізу вершин графу, що мають багато попередників, у десятки та навіть сотні разів. Відбувається це завдяки виключенню операцій над елементами множини та збільшенню пам'яті для зберігання матриці ідентифікаторів вершин. Треба зазначити, що час пошуку ЗР при роботі препроцесору майже не відрізняється від часу моделювання дедуктивним або дедуктивно-паралельним методом на вхідному наборі.

Запропонований дедуктивний аналіз та синтез дедуктивних компонентів для SoC. Дедуктивний метод моделювання несправностей об'єднує в собі переваги раніше розглянутих в цьому розділі методів: дедуктивного виявлення переліків несправностей, що є ефективним с точки зору математики, та виконання паралельних процедур, що орієнтовані на обробку з високою швидкістю ЦП вентильного, регістрового та системного рівнів опису.

Усі запропоновані методи призначені для зниження трудомісткості, скорочення часових витрат і підвищення ефективності процедури пошуку несправностей у цифрових системах, в тому числі у цифрових системах на кристалах.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної магістерської роботи було проведення дослідження методів вдосконалення алгоритмів пошуку дефектів у цифрових системах. Головною задачею дослідження є аналіз існуючих засобів та методів пошуку дефектів у цифрових системах на кристалах, пошук їх головних недоліків та розробка нового алгоритму і/або методу тестування, що буде досконаліший попередників таких як послідовний метод діагностики несправностей, умовний алгоритм пошуку дефектів за допомогою побудови дерева, автоматичні системи тестування фірм Logic Vision, Synopsys, Mentor Graphics, Cadence. За цим в подальшому буде запропонований реальний алгоритм, що значно полегшить пошук дефектів у цифрових системах на кристалах. Так як в процесі проектування використовувалося комп'ютерне програмне забезпечення, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для робочого місця з використанням персонального комп'ютера на якому буде розроблятися/використовуватися розроблена програма.

5.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» [52] визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів ший, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших

засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці.

5.2 Правові та організаційні основи охорони праці

Державна політика в галузі охорони праці визначається відповідно до Конституції України Верховною Радою України і спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Відповідно до статті 3 Закону України "Про охорону праці" (далі – Закону) [52] законодавство про охорону праці складається з Закону, Кодексу законів про працю України, Закону України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності"[53] та прийнятих відповідно до них нормативно-правових актів, норм міжнародного договору (ратифіковані Конвенції і Рекомендації МОТ, директиви Європейської Ради).

5.3 Аналіз стану умов праці

Робота над створенням автоматизованої системи проходитиме в приміщенні багатоквартирного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

5.3.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 5.1.

Таблиця 5.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	3
Висота, м	3
Площа, м ²	15
Об'єм, м ³	45

Згідно з [54] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для забезпечення потрібного рівного освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі.

5.3.2 Вимоги до організації місця праці

Робочий стіл на досліджуваному місці також містить достатньо простору для ніг. Крісло, що використовується в якості робочого сидіння, є підйомно поворотним, має підлокітники і можливість регулювання за висотою і кутом нахилу спинки, також воно м'яке і виконане з екологічної шкіри, що дає можливість працювати у комфорті. Екран монітору знаходиться на відстані 0.8 м, клавіатура має можливість регулювання кута нахилу 5-15°. Отже, за всіма параметрами робоче місце відповідає нормативним вимогам. Приміщення кабінету знаходиться на третьому поверсі дев'яти поверхової будівлі і має об'єм 45 м³, площу – 15 м². У цьому кабінеті обладнано одне місце праці, з одним укомплектованим ПК.

Температура в приміщенні протягом року коливається у межах 18–24°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум в лабораторії знаходиться на рівні 50 дБА. Система вентиляції приміщення — природна неорганізована, а опалення — централізоване.

Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою восьми люмінесцентних ламп, забезпечує рівень освітленості не менше 200 Лк.

У приміщенні є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним струмом.

Наявна аптечка для надання долікарської допомоги, а також у приміщенні роблять вологе прибирання та щоденно провітрюють.

5.3.3 Навантаження та напруженість процесу праці

Під час виконання магістерської роботи: за фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки

певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Під час виконання робіт використовують ПК, що призводить до навантаження на окремі системи організму. Зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Тобто наявне психофізіологічні небезпечні та шкідливі фактори:

а) фізичного перевантаження:

- 1) статичного;
- 2) динамічного;

б) нервово-психічного перевантаження:

- 1) розумового перенапруження;
- 2) монотонності праці;
- 3) перенапруження аналізаторів;
- 4) емоційних перевантажень.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи наведені в [55].

Роботу за дипломним проектом визнано, таку, що займає 70% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви для дослідника тривалістю 15 хв через кожну годину роботи.

5.4 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

5.4.1 Аналіз небезпечних та шкідливих факторів при роботі з персональним комп'ютером

Робота виконувалася у приміщенні з персональним комп'ютером (ПК) та периферійними пристроями. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга $U = +220\text{В} \pm 5\%$;
- робочий струм $I = 2\text{А}$;
- споживана потужність $P = 350\text{Вт}$.

Робочі місця мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [55]. За умов роботи з ПК виникають наступні небезпечні та шкідливі чинники: несприятливі мікрокліматичні умови, освітлення, електромагнітні випромінювання, шум, вібрація, електричний струм, електростатичне поле, напруженість трудового процесу та інше.

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 5.2).

Таблиця 5.2 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
1	2	3	4
психофізіологічні:			
- нервово-психічне перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - моделювання та аналіз алгоритмів; - написання коду, тестування; - оформлення роботи	5	[56, 57]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці- сидіння користувача,) та організації робочого часу - безпервна робота)	2	

Продовження таблиці 5.2

<i>фізичні</i>			
- підвищена температура поверхонь обладнання	експлуатація ноутбуку	2	[54]
недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[58]
недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[58]
- підвищена яскравість світла	порушення умов праці (організації місця праці- налагодження моніторів)	1	[55]
- підвищений рівень вібрації	-//-	1	[59, 60]
- підвищений рівень електромагнітного випромінення	-//-	2	[61]

5.4.2 Пожежна безпека

Небезпека розвитку пожежі на робочому місці пов'язана з особливістю комп'ютерів - із значною кількістю щільно розташованих на монтажній платі і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Надійна робота окремих елементів і мікросхем в цілому забезпечується тільки в певних інтервалах температури, вологості і при заданих електричних параметрах. При відхиленні реальних умов експлуатації від розрахункових можуть виникнути пожежонебезпечні ситуації.

Кабельні лінії є найбільш пожежонебезпечними місцем. Наявність пального ізоляційного матеріалу, ймовірних джерел запалювання у вигляді електричних іскор і дуг, розгалуженість і недоступність роблять кабельні лінії місцем найбільш ймовірного виникнення і розвитку пожежі. Для зниження займистості і здатності поширювати полум'я кабелі покривають вогнезахисними покриттями.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно

захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу.

5.4.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК та периферійні пристрої, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три-провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

5.5 Гігієнічні вимоги до параметрів виробничого середовища

5.5.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт Ia. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [54].

Дане приміщення обладнане системами опалення та припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря у відповідності до [54].

Для забезпечення оптимальних параметрів мікроклімату в приміщенні проводяться перерви в роботі співробітників, з метою його провітрювання.

5.5.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

Освітленість приміщення має велике значення при роботі на ПЕОМ. Вона багато в чому визначається колірною і мережевий обстановкою. Для зменшеного поглинання світла стеля і стіни вище панелей (1,5-1,7м.). Якщо вони не облицьовані звукопоглинальним матеріалом, фарбуються білою водоемульсійною фарбою (коефіцієнт відбиття повинен бути не менше 0,7). Для забарвлення стіни панелей рекомендується віддавати перевагу світлим фарбам.

У проєкті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (75 Вт), тривалий термін служби (до 10000 годин), спектральний складом випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IV в розряді точності (середня точність). При цьому нормована освітленість на робочому місці (E_n) рівна 200 лк. Джерелом природного освітлення є сонячне світло.

У приміщенні, де розташовані ЕОМ природне бічне освітлення, рівень якого відповідає стандарту [58].

Розрахунок освітлення.

Для виробничих будівель світловий коефіцієнт приймається в межах 1/6 - 1/10:

$$\sqrt{a^2 + b^2} \cdot S_b = (1/8 \div 1/10) \cdot S_n \quad (5.1)$$

де S_b – площа віконних прорізів, m^2 ;

S_n – площа підлоги, m^2 .

$S_n = a \cdot b = 3 \cdot 5 = 15 m^2$,

$$S = 1/10 \cdot 15 = 1,5 \text{ м}^2.$$

Приймаємо 2 вікна площею $S=1,6 \text{ м}^2$ кожне.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 75 Вт) з світловим потоком 5200 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі (5.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (5.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, м^2 ; $S = 15 \text{ м}^2$;

Z – поправочний коефіцієнт світильника ($Z = 1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,15;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число ламп розжарювання в світильнику – 1;

F – світловий потік лампи – 5200лм (для ЛБ-75).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 * 15 * 1,15 * 1,5}{5200 * 0,575 * 1} = \frac{7762,5}{2990} = 2,596 \approx 2,6$$

Приймаємо освітлювальну установку, яка складається з одного світильника з однією люмінесцентною лампою потужністю 75 Вт, напругою – 220 В.

5.6 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення $> 45 \text{ м}^3$ на одного

працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП (30 м³ на годину на одного працюючого).

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

5.7 Охорона навколишнього природного середовища

Вплив виробничих факторів на навколишнє природне середовище регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища» [62], Законом України «Про забезпечення санітарного та епідемічного благополуччя населення» [63], Законом України «Про відходи» [64], Законом України «Про охорону атмосферного повітря» [65], Законом України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру» [66], Водний кодекс України [67].

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на видалення (знешкодження), утилізацію, тощо в ІТ галузі.

Вплив на атмосферне повітря при нормальних умовах праці не оказує, бо не має в приміщенні сканерів, принтерів та інших джерел викиду забруднюючих речовин в повітря робочої зони.

В процесі дослідження методів вдосконалення алгоритмів пошуку дефектів у цифрових системах виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- відпрацьовані люмінесцентні лампи - I клас небезпеки;
- акумулятор для джерел безперебійного харчування -III клас небезпеки;
- змінні носії інформації - IV клас небезпеки;
- побутові відходи - IV клас небезпеки.

5.8 Висновки до розділу 5

В даному розділі проведено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в магістерській дипломній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

Визначені основні екологічні аспекти впливу на навколишнє природне середовище та зазначені заходи щодо поводження з відходами виробничої діяльності.

ВИСНОВКИ

В **першому** розділі магістерської роботи здійснено огляд методів і засобів пошуку дефектів у цифрових системах. Визначені такі поняття, як технічна діагностика, надійність, справність, працездатність, правильне функціонування, дефект, несправність, контроль технічного стану, алгоритм технічного діагностування.

Розглянуті основи технічного та тестового діагностування. Проведений аналіз існуючих несправностей в цифрових системах.

Проведений огляд та аналіз існуючих систем тестування цифрових систем на кристалах. Зроблений висновок щодо недоліків даних систем, на основі яких базується мета магістерської роботи.

Метою магістерської роботи є дослідження методів та засобів вдосконалення алгоритмів пошуку дефектів у цифрових системах за допомогою дедуктивних та дедуктивно-паралельних методів та розроблення рекомендацій щодо застосування розглянутих методів..

У **другому** розділі магістерської роботи розглянуто класифікацію та побудову алгоритмів діагностування, що використовують таблиці несправностей. Визначені такі поняття, як перелік несправностей, константна несправність, класи еквівалентності та довжина алгоритму діагнозу. Зазначено, що безпосереднє використання таблиці функцій несправностей, як форми подання інформації при побудові і реалізації алгоритмів діагностики і фізичних моделей об'єктів, часто неможливо через високу розмірність таблиці.

Однак як універсальна математична модель об'єкта діагностики таблиця функцій несправностей дуже наочна і зручна при обговоренні та класифікації принципів, а також основних процедур побудови та реалізації алгоритмів діагностики, навіть якщо ці принципи і процедури спочатку формулюються на мовах, відмінних від мови таблиць функцій несправностей. Визначені властивості таблиць функцій несправності. Запропоновано використання умовного алгоритму пошуку дефектів у вигляді дерева, яке будується на основі таблиці несправності, для якої дотримуються три умови - виявлення, розрізнення і мінімальність. В даному алгоритмі пошуку дефекту в якості елементарної перевірки P_i розглядається подача на схему одного двійкового набору і порівняння реакцій схеми в контрольних точках (на зовнішніх виходах) з еталоном.

У **третьому** розділі магістерської роботи розглянуто структурні алгоритми пошуку дефектів такі, як методи стиску двійкових послідовностей, зондовий алгоритм пошуку дефектів, зворотнє простеження та умовний зондовий алгоритм пошуку дефектів. Визначені такі поняття,

як тест, сигнатура, об'єкт діагностування, зондова елементарна перевірка, алгоритм діагностування, контрольна точка, наступники та діагностичний експеримент.

Зазначено, що сигнатурний аналіз дозволяє з високою точністю локалізувати несправність і не вимагає високої кваліфікації обслуговуючого персоналу, однак його недоліком є непридатність для захисту від нових вірусів.

Алгоритм діагностування за допомогою зонду не диференціюється чітко на процедури контролю та пошуку дефектів та передбачає напівавтоматичний діалоговий режим перевірки за участю інженера-оператора, що є дуже ефективним способом діагностики. Однак недоліком зондової організації перевірок є необхідність забезпечити хороший контакт в місці з'єднання зонду і досліджуваного елемента.

Визначені властивості матриць досяжності та дерев пошуку несправностей.

Запропоновано використання методів стиску двійкових послідовностей, зондового алгоритму пошуку дефектів, зворотного простеження та умовного зондового алгоритму пошуку дефектів. Для кожного методу наведено приклади їх використання на практиці.

У **четвертому** розділі магістерської роботи розглянутий паралельно-дедуктивний метод виявлення дефектів у цифрових системах. Розглянута узагальнена модель дедуктивно-паралельного аналізу несправностей, у якій був запропонований алгоритм, орієнтований як на табличний опис примітивів довільної складності RTL-рівня, так і на вентильне представлення цифрових схем. Швидкодія алгоритму інваріантна компілятивним та інтерпретативним моделям цифрових пристроїв. Однак з позиції програмування чисто інтерпретативна реалізація є найбільш технологічною.

Запропонований дедуктивний метод структурного аналізу, дедуктивно-паралельний метод моделювання графових структур. Головними перевагами модифікованих для дедуктивного методу структурного аналізу формул є те, що операції над вершинами графу виконуються паралельно при програмному препроцесорі моделювання. Це підвищує швидкодію аналізу вершин графу, що мають багато попередників, у десятки та навіть сотні разів. Відбувається це завдяки виключенню операцій над елементами множини та збільшенню пам'яті для зберігання матриці ідентифікаторів вершин. Треба зазначити, що час пошуку ЗР при роботі препроцесору майже не відрізняється від часу моделювання дедуктивним або дедуктивно-паралельним методом на вхідному наборі.

Запропонований дедуктивний аналіз та синтез дедуктивних компонентів для SoC. Дедуктивний метод моделювання несправностей об'єднує в собі переваги раніше розглянутих в цьому розділі методів: дедуктивного виявлення переліків несправностей, що є ефективним с точки зору математики, та виконання паралельних процедур, що орієнтовані на обробку з високою швидкістю ЦП вентильного, реєстрового та системного рівнів опису.

Усі запропоновані методи призначені для зниження трудомісткості, скорочення часових витрат і підвищення ефективності процедури пошуку несправностей у цифрових системах, в тому числі у цифрових системах на кристалах.

У **п'ятому** розділі магістерської роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в магістерській дипломній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

А також визначені основні екологічні аспекти впливу на навколишнє природне середовище та зазначені заходи щодо поводження з ними.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ziad Noun. Wireless Approach for SIP and SOC Testing. 2010. URL: https://tel.archives-ouvertes.fr/file/index/docid/512832/filename/Ziad_Noun_-_Wireless_Approach_for_SiP_and_SoC_testing.pdf (дата звернення: 25.08.2018).
2. Литвинова Евгения Ивановна Технологии встроенного тестирования system-in-package // Радиоэлектроника и информатика. 2008. №3. URL: <https://cyberleninka.ru/article/n/tehnologii-vstroennogo-testirovaniya-system-in-package> (дата звернення: 25.08.2018).
3. Касабуцька І.В., Кардашук В.С. Методи вдосконалення тестування цифрових систем на кристалах// VIII Всеукраїнська науково-практична конференція «Електронні апарати та системи. Проблеми створення. Перспективи розвитку»: матеріали VII Всеукр.науково-практ. конф., м. Сєверодонецьк. / М-во освіти і науки України, Східноукр. Нац. ун-т ім. В.Даля; редкол.: Поркунян О.В. [та ін.] – Сєверодонецьк: Східноукр. нац. ун-т ім. В. Даля, 2018– 239 с.
4. Касабуцька І.В., Кардашук В.С. Методи вдосконалення тестування цифрових систем. Вісник Східноукраїнського національного університету ім. В. Даля. 2018. №.6 (247). С. 61-68. (фахове видання).
5. Технічне діагностування радіоелектронних пристроїв. URL: https://uk.wikipedia.org/wiki/Технічне_діагностування_радіоелектронних_пристроїв (дата звернення: 25.08.18).
6. Карчевський В. П. Функціональне діагностування пристроїв з елементів, які здійснюють генерування та трансляцію сигналів про відмови. – 2009.URL: <http://dspace.nbuu.gov.ua/bitstream/handle/123456789/7877/02-Karchevskiy.pdf?sequence=1> (дата звернення: 25.08.18).
7. Тестове діагностування цифрових пристроїв. URL: https://uk.wikipedia.org/wiki/Тестове_діагностування_цифрових_пристроїв (дата звернення: 25.08.18).
8. Засоби діагностування та їх класифікація. URL: https://studopedia.su/2_41178_zasobi-diaagnostuvannya-ta-ih-klasifikatsiya.html (дата звернення: 25.08.18).
9. Ma T. P., Dressendorfer P. V. (ed.). Ionizing radiation effects in MOS devices and circuits. – John Wiley & Sons, 1989.
10. Григорьев А. В., Данилова Е. А., Држевецкий А. Л. Классификация дефектов бортовой РЭА // НиКа. 2013. №. URL: <https://cyberleninka.ru/article/n/klassifikatsiya-defektov-bortovoy-rea> (дата звернення: 09.10.2018).
11. М. Ф. Каравай. Алгоритм построения проверяющего теста для кратных неисправностей по структуре комбинационного устройства. Автомат. и телемех. 1975. №1. 162–

170. URL: http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=at&paperid=7778&option_lang=rus (дата звернения: 25.08.2018).

12. С. Й. Олиховский, М. М. Белова, Е. В. Кочелаб. Кинетика образования и роста микродефектов в кристаллах // Успехи физ. мет. 2006 URL: https://www.researchgate.net/profile/S_Olikhovskii/publication/266736698_Kinetics_of_nucleation_and_growth_of_microdefects_in_crystals_in_Russian/links/5439a20f0cf2d6698be1676a/Kinetics-of-nucleation-and-growth-of-microdefects-in-crystals-in-Russian.pdf (дата звернения: 25.08.2018).

13. Jha N. K., Gupta S. Testing of digital systems. – Cambridge University Press, 2003.

14. Blanks H. S. The temperature dependence of component failure rate //Microelectronics Reliability. – 1980. – Т. 20. – №. 3. – С. 297-307.

15. Harman G. G. Wire bonding in microelectronics. – McGraw-Hill, 2010.

16. Carter P. M., Wilkins B. R. Influences on soft error rates in static RAMs //IEEE Journal of Solid-State Circuits. – 1987. – Т. 22. – №. 3. – С. 430-436.

17. Carter P. M., Wilkins B. R. Alpha particle induced soft errors in NMOS RAMs: a review //IEE Proceedings I-Solid-State and Electron Devices. – 1987. – Т. 134. – №. 1. – С. 32-44.

18. Nadeau-Dostie B. (ed.). Design for at-speed test, diagnosis and measurement. – Springer Science & Business Media, 2000. – Т. 15.

19. Слинкин Д.И. Анализ современных методов тестирования и верификации проектов сверхбольших интегральных схем // Программные продукты и системы. 2017. №3. URL: <https://cyberleninka.ru/article/n/analiz-sovremennyh-metodov-testirovaniya-i-verifikatsii-proektov-sverhbolshih-integralnyh-shem> (дата обращения: 25.10.2018).

20. Electronic design automation. URL: https://en.wikipedia.org/wiki/Electronic_design_automation (дата звернения: 25.08.18).

21. Design for testing. URL: https://en.wikipedia.org/wiki/Design_for_testing (дата звернения: 25.08.18).

22. Built-in self-test. URL: https://en.wikipedia.org/wiki/Built-in_self-test (дата звернения: 25.08.18).

23. The Wall Street Transcript. 2003-11-10. Retrieved 2018-01-04. URL: <https://www.twst.com/interview/vinod-agarwal-logicvision-inc-1gyn> (дата звернения: 25.08.18).

24. *Portland Business Journal*. August 18, 2009. Retrieved 2009-08-22.

25 Savas Z., Arik M. ATE hardware independent TPS development using a reconfigurable test executive //Proc. Autotestcon 2013. – 2013.


26. Synopsys. URL: <https://www.synopsys.com/> (дата звернения: 25.08.18).

27. Bhatnagar H. Advanced ASIC Chip Synthesis: Using Synopsys® Design Compiler™ Physical Compiler™ and PrimeTime®. – Springer Science & Business Media, 2007.

28. Долинский Михаил "горячие темы" еда индустрии по материалам новостей портала dacafe. Com // Компоненты и Технологии. 2004. №7. URL: https://kit-e.ru/assets/files/pdf/2004_07_146.pdf (дата звернення: 10.10.2018).
29. Mentor Graphics. URL: https://en.wikipedia.org/wiki/Mentor_Graphics (дата звернення: 25.08.18).
30. Logic Test - Structural test using. URL: https://www.mentor.com/products/silicon-yield/logic_test/ (дата звернення: 25.08.18).
31. The Tessent® Product Suite/ URL: <https://www.mentor.com/products/silicon-yield/tessent/> (дата звернення: 25.08.18).
32. Memory Test. URL: https://www.mentor.com/products/silicon-yield/memory_test/ (дата звернення: 25.08.18).
33. Mixed Signal Test. URL: <https://www.mentor.com/products/silicon-yield/mixed-signal-test> (дата звернення: 25.08.18).
34. SerDes. URL: <https://ru.wikipedia.org/wiki/SerDes> (дата звернення: 25.08.18).
35. Silicon Learning/ URL: <https://www.mentor.com/products/silicon-yield/silicon-learning/> (дата звернення: 25.08.18).
36. Cadence Design Systems. URL: https://ru.wikipedia.org/wiki/Cadence_Design_Systems (дата звернення: 25.08.18).
37. Integrated Circuits and Devices for PMICs. URL: <http://ecee.colorado.edu/~ecen5007/software.html> (дата звернення: 25.08.18).
38. Automatic Placement and Routing using Cadence Encounter//6.375 Tutorial 5 March 16, 2006. URL: http://csg.csail.mit.edu/6.375/6_375_2006_www/handouts/tutorials/tut5-enc.pdf (дата звернення: 25.10.2018).
39. Department of Electronics. URL: http://www.doe.carleton.ca/facilities/computer_help/internal/software.php (дата звернення: 25.08.18).
40. Appello D., Bernardi P., Grosso M., Reorda M.S. System-in-package testing: problems and solutions // IEEE Design & Test of Computers. Vol. 23, Issue 3. May-June, 2006. P. 203 – 211.
41. Городецкий Ами Встроенные инструменты тестирования // Компоненты и Технологии. 2009. №92. URL: <https://cyberleninka.ru/article/n/vstroennye-instrumenty-testirovaniya> (дата звернення: 23.08.2018).
42. Городецкий Ами Новый JTAG-стандарт IEEE 1149. 7 // Компоненты и Технологии. 2010. №105. URL: <https://cyberleninka.ru/article/n/novyy-jtag-standart-ieee-1149-7> (дата звернення: 23.08.2018).


43. Farrokh Ghani Zadegan , Urban Ingelsson , Gunnar Carlsson and Erik Larsson. Design Automation for IEEE P1687. 2011. URL: https://www.ida.liu.se/labs/eslab/publications/pap/db/farza_DATE11.pdf (дата звернення: 23.08.2018).
44. IEEE-SA Standards Board: веб-сайт. URL: <http://grouper.ieee.org/groups/1687/> (дата звернення: 23.08.2018).
45. Subhasish M.\$ Huang W.-J., Saxena N.R., Yu S.-Y., McCluskey E.J. Reconfigurable architecture for autonomous self-repair // IEEE Design & Test of Computers. Volume 21, Issue 3. May-June, 2004. P.228-240.
46. Ross R., Hall R. A FPGA Simulation Using Asexual Genetic Algorithms for Integrated Self-Repair // Adaptive Hardware and Systems, 2006.-AHS 2006. First NASA/ESA Conference on Volume. Issue 15-18 June 2006. P. 301-304
47. Habermann S., Kothe R., Vierhaus H.T. Built-in self repair by reconfiguration of FPGAs // Proceeding of the 12th IEEE International Symposium on On-Line Testing. 2006. P. 187-188.
48. Pontarelli, M. Ottavi, V. Vankamamidi, A. Salsano, F Lombardi. Reliability Evaluation of Repairable/Reconfigurable FPGAs // 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06)/ October, 2006. P. 227-235.
49. Автоматизированное проектирование цифровых устройств / С.С. Бадулин, Ю.М. Барнаулов и др./ Под ред. С.С. Бадулина. М.: Радио и связь, 1981. 240 с.
- 50 Hahanov V.I, Babich A.V., Hyduke S.M. Test Generation and Fault Simulation Methods on the Basis of Cubic Algebra for Digital Devices // Proceeding of the Euromicro Symposium on Digital System Design DSD2001/ Warsaw, Poland. September, 4-6, 2001. P. 228-235.
51. Abramovici M., Breuer M.A. and Friedman A.D. Digital systems testing and testable design. Computer Science Press. 1998. 652 p.
52. Закон України "Про охорону праці". URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 25.10.2018).
53. Закон України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності". URL: <https://zakon.rada.gov.ua/laws/show/1105-14/ed20140401#o90> (дата звернення: 25.10.2018).
54. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень».
55. Постанова Головного державного санітарного лікаря України від 10.12.98 № 7 ДСанПіН 3.3.2-007-98..
56. НПАОП 0.00-1.28-10 "Про затвердження правил охорони праці під час експлуатації електронно-обчислювальних машин".
57. ДБН В.2.5-28:2015 "Природне і штучне освітлення".

58. ДСН 3.3.6.037-99 "Санітарні норми виробничого шуму, ультразвуку та інфразвуку".
59. ДСТУ ГОСТ 12.1.012-90 ССБТ. Вибрационная безопасность. Общие требования.
60. ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности. Допустимые уровни на рабочих местах и требования к проведению контроля.
61. Закон України «Про охорону навколишнього природного середовища». URL: <http://zakon.rada.gov.ua/laws/show/1264-12> (дата звернення: 24.12.2018).
62. Закон України «Про забезпечення санітарного та епідемічного благополуччя населення». URL: <https://zakon.rada.gov.ua/laws/show/4004-12> (дата звернення: 24.12.2018).
63. Закон України «Про відходи» . URL: <https://zakon.rada.gov.ua/laws/show/187/98-%D0%B2%D1%80>.
64. Закон України «Про охорону атмосферного повітря». URL: <https://zakon.rada.gov.ua/laws/show/2707-12> (дата звернення: 24.12.2018).
65. Закон України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру». URL: <https://zakon.rada.gov.ua/laws/show/1809-14> (дата звернення: 24.12.2018).
66. Водний кодекс України. URL: <https://zakon.rada.gov.ua/laws/show/213/95-%D0%B2%D1%80> (дата звернення: 24.12.2018).



ПОСТАНОВКА ЗАДАЧІ

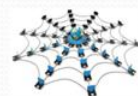
- Метою магістерської роботи є дослідження методів та засобів вдосконалення алгоритмів пошуку дефектів у цифрових системах за допомогою дедуктивних та дедуктивно-паралельних методів.
- Головною задачею дослідження є аналіз існуючих засобів та методів пошуку дефектів у цифрових системах на кристалах, пошук їх головних недоліків та розробка нового алгоритму і/або методу тестування, що буде досконаліший попередників



Стан питання

- Домінування цифрових систем на кристалах (SoC) та цифрових систем в пакетах (SiP) пояснюється тим, що розвиток високих технологій дозволив об'єднати цифрові, аналогові, радіочастотні та навіть компоненти мікроелектромеханічних систем (MEMS) в одному пристрої – чіпі, що зменшує загальний розмір системи та підвищує її продуктивність. На сьогоднішній день створення мініатюрних, спеціалізованих, економічних мобільних та швидкодіючих цифрових виробів є дуже затребуваним у всьому світі.

Розглянуті такі існуючі
системи тестування як:



SYNOPSYS®

Mentor
Graphics® cādence

5



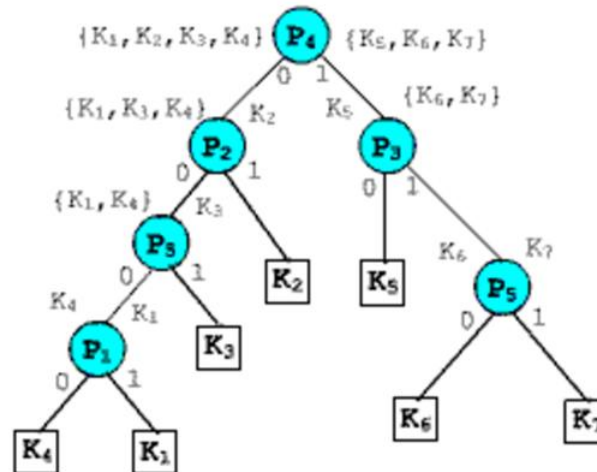
Шляхи вирішення задачі

- Класифікація та побудова алгоритмів діагностування;
- Структурні алгоритми пошуку дефектів в системах діагностування;
- Використання дедуктивно-паралельного методу для виявлення несправностей у цифрових системах



Класифікація та побудова алгоритмів діагностування

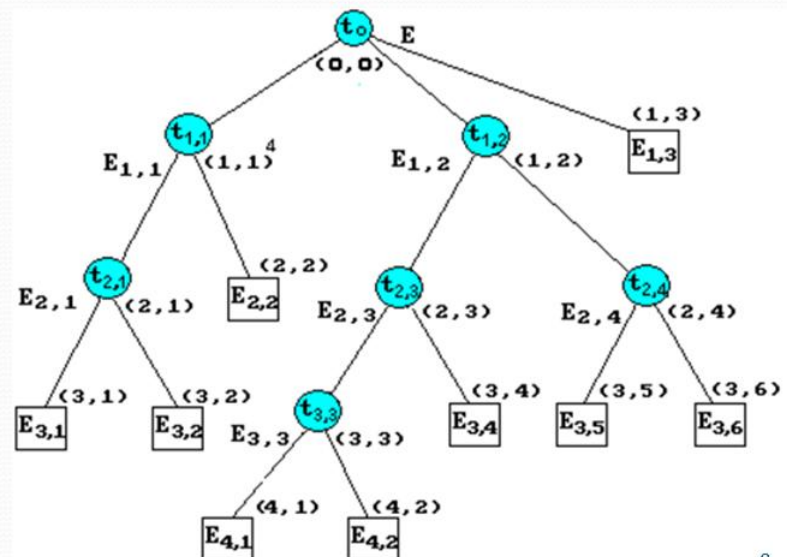
- Умовний алгоритму пошуку дефектів у вигляді дерева



7

Класифікація та побудова алгоритмів діагностування

- Алгоритми діагностики



8

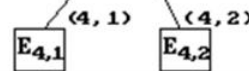
Класифікація та побудова алгоритмів діагностування

- Алгоритми діагностики



Алгоритм діагностики задає сукупність елементарних перевірок, послідовність (або послідовності) їх реалізації та правила обробки результатів, що реалізуються елементарними перевірками з метою отримання результатів діагностики.

6)

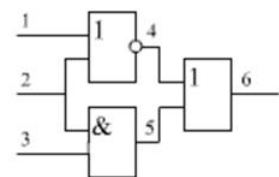


8

Класифікація та побудова алгоритмів діагностування

- Розглянуто класифікацію та побудову алгоритмів діагностування, що використовують таблиці несправностей. Зазначено, що безпосереднє використання таблиці функцій несправностей, як форми подання інформації при побудові і реалізації алгоритмів діагностики і фізичних моделей об'єктів, часто неможливо через високу розмірність таблиці.

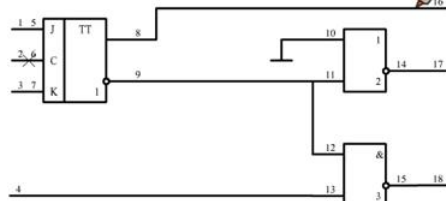
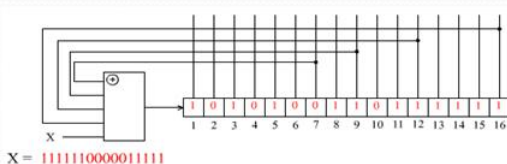
R	E		
	e	\dots	$e_{[s]}$
π_1	R_1		$R_1^{[s]}$
\vdots			
π_j	R_j		$R_j^{[s]}$
\vdots			
$\pi_{[n]}$	$R_{[n]}$		$R_{[n]}^{[s]}$



9

Структурні алгоритми пошуку дефектів в системах діагностування

- Методи стиску двійкових послідовностей. Сигнатурний аналіз;
- Зондовий алгоритм пошуку дефектів. Зворотне простеження;
- Умовний зондовий алгоритм пошуку дефекту



П+ П- Sign
17 08 9F3A

10

Структурні алгоритми пошуку дефектів в системах діагностування



- Сигнатурний аналіз дозволяє з високою точністю локалізувати несправність і не вимагає високої кваліфікації обслуговуючого персоналу, однак його недоліком є непридатність для захисту від нових вірусів.
- Алгоритм діагностування за допомогою зонду не диференціюється чітко на процедури контролю та пошуку дефектів та передбачає напівавтоматичний діалоговий режим перевірки за участю інженера-оператора, що є дуже ефективним способом діагностики. Однак недоліком зондової організації перевірок є необхідність забезпечити хороший контакт в місці з'єднання зонду і досліджуваного елемента.

11

ДЕДУКТИВНО-ПАРАЛЕЛЬНИЙ МЕТОД ДЛЯ ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ У ЦИФРОВИХ СИСТЕМАХ

- Узагальнена модель дедуктивно-паралельного методу аналізу несправностей

X1	X2	Y1
0	1	0

 \oplus

X1	X2	Y1
0	0	0
0	1	0
1	0	1
1	1	1

 $=$

X1	X2	Y1
0	1	0
0	0	0
1	1	1
1	0	1

$$L_1 = X_1X_2 \cup X_1\bar{X}_2$$

X1	X2	Y1
1	1	1

 \oplus

X1	X2	Y1
0	0	0
0	1	0
1	0	1
1	1	1

 $=$

X1	X2	Y1
1	1	1
1	0	1
0	1	1
0	0	0

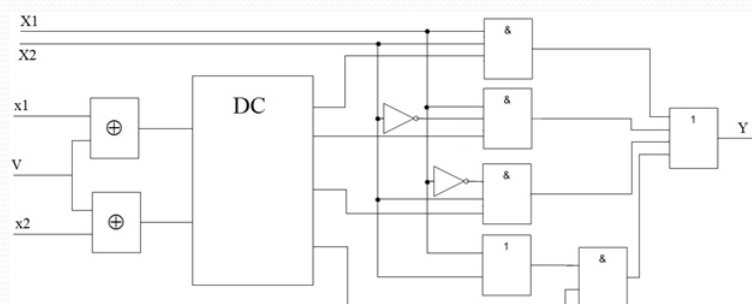
$$L_2 = X_1X_2 \cup X_1\bar{X}_2 \cup \bar{X}_1X_2$$

12

ДЕДУКТИВНО-ПАРАЛЕЛЬНИЙ МЕТОД ДЛЯ ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ У ЦИФРОВИХ СИСТЕМАХ

- Узагальнена модель дедуктивно-паралельного методу аналізу несправностей

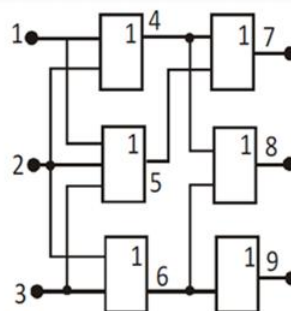
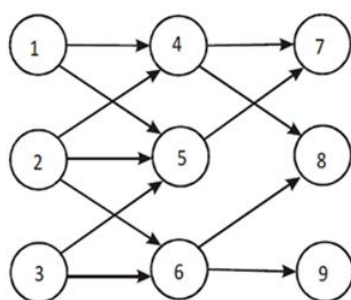
Апаратна реалізація
універсального
функціонального
примітиву
(симулятора)
дедуктивно-
паралельного
аналізу
несправностей



13

ДЕДУКТИВНО-ПАРАЛЕЛЬНИЙ МЕТОД ДЛЯ ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ У ЦИФРОВИХ СИСТЕМАХ

- Дедуктивний і дедуктивно-паралельний методи структурного аналізу схеми та моделювання графових структур



14

ДЕДУКТИВНО-ПАРАЛЕЛЬНИЙ МЕТОД ДЛЯ ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ У ЦИФРОВИХ СИСТЕМАХ

- Дедуктивний аналіз та синтез дедуктивних компонентів для SoC

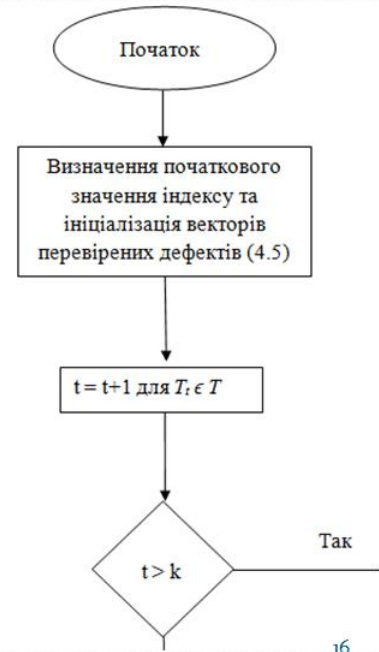
Таблиця 4.8 – Карта Карно D-тригера

(T)\(X)	000	001	011	010	110	111	101	100
000	0	1	1	0	1	1	0	0
001	0	1	1	0	0	0	1	1
011	0	1	1	0	1	1	0	0
010	0	1	1	0	0	0	1	1
110	0	0	1	1	1	0	0	1
111	0	0	1	1	0	1	1	0
101	0	0	1	1	1	0	0	1
100	0	0	1	1	0	1	1	0

15

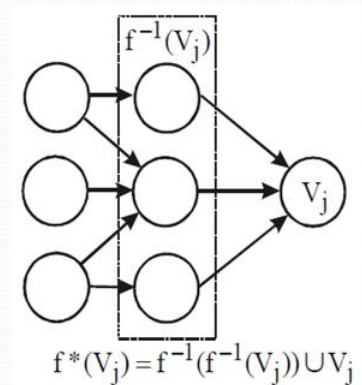
Отримані результати та висновки

- Розглянута узагальнена модель дедуктивно-паралельного аналізу несправностей, у якій був запропонований алгоритм, орієнтований як на табличний опис примітивів довільної складності RTL-рівня, так і на вентильне представлення цифрових схем. Швидкодія алгоритму інваріантна компілятивним та інтерпретативним моделям цифрових пристроїв. Однак з позиції програмування чисто інтерпретативна реалізація є найбільш технологічною.



Отримані результати та висновки

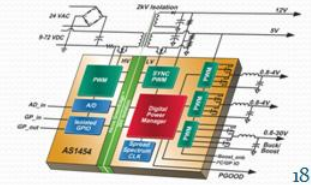
- Запропонований дедуктивний метод структурного аналізу, дедуктивно-паралельний метод моделювання графових структур. Операції над вершинами графу виконуються паралельно при програмному препроцесорі моделювання. Це підвищує швидкодію аналізу вершин графу, що мають багато попередників, у десятки та навіть сотні разів.



Отримані результати та висновки



- Запропонований дедуктивний аналіз та синтез дедуктивних компонентів для SoC. Дедуктивний метод моделювання несправностей об'єднує в собі переваги раніше розглянутих в цьому розділі методів: дедуктивного виявлення переліків несправностей, що є ефективним з точки зору математики, та виконання паралельних процедур, що орієнтовані на обробку з високою швидкістю ЦП вентильного, регістрового та системного рівнів опису.



18

Отримані результати та висновки

- Усі запропоновані методи призначені для зниження трудомісткості, скорочення часових витрат і підвищення ефективності процедури пошуку несправностей у цифрових системах, в тому числі у цифрових системах на кристалах.



19