

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)  
(рівень вищої освіти)

МЕТОДИ ДІАГНОСТУВАННЯ ДЕФЕКТІВ У КІБЕРФІЗИЧНИХ  
СИСТЕМАХ НА ОСНОВІ МАТРИЧНИХ СТРУКТУР ДАНИХ  
(тема)

Виконав: студент II курсу, групи СКСм-22-1

Каряка Е.В.  
(прізвище, ініціали)

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

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

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

Керівник проф. Хаханов В.І.

Допускається до захисту

Зав. каф. АПОТ



(підпис)  
2023 р.

Чумаченко С.В.  
(прізвище, ініціали)

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:  
Зав. кафедри АПОТ



Чумаченко С.В.  
(підпис)

«03» листопада 2023 р.

**ЗАВДАННЯ**  
НА КВАЛІФАКАЦІЙНУ РОБОТУ

студентові Каряка Ернесту Васильовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Методи діагностування дефектів у кіберфізичних системах на основі матричних структур даних

затверджена наказом по університету від 03 листопада 2023 р. № 1282 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19 січня 2024 р.

3. Вихідні дані до роботи \_\_\_\_\_

Методи діагностування дефектів

Елементи булевої логіки

4. Перелік питань, що потрібно опрацювати у роботі \_\_\_\_\_

Аналіз предметної області

Метрики та виміри у кіберфізичному просторі

Апаратна реалізація секвенсора

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) \_\_\_\_\_  
слайди презентації – 18 \_\_\_\_\_  
\_\_\_\_\_

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

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів	Примітка
1	Отримання завдання	01.09.2023-05.09.2023	
2	Аналіз предметної області, тенденції розвитку цифрових систем на кристалах	07.09.2023-21.09.2023	
3	Метрики та виміри у кіберфізичному просторі	22.09.2023-15.10.2023	
4	Аналіз математичного апарату – методів діагностування дефектів	16.10.2023-06.11.2023	
5	Апаратна реалізація секвенсора	07.11.2023-07.12.2023	
6	Оформлення пояснювальної записки	08.12.2023-30.12.2023	
7	Оформлення графічного матеріалу	02.01.2024-05.01.2024	
8	Перевірка виконаного проекту керівником	06.01.2024-12.01.2024	

Дата видачі завдання 01 вересня 2023 р. \_\_\_\_\_

Студент \_\_\_\_\_ Каряка Е.В. \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Хаханов В.І. \_\_\_\_\_  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 56 с., 25 рис., 2 табл., 22 джерела.

МОДЕЛЬ, МЕТОД, АРХІТЕКТУРА, КІБЕРФІЗИЧНИЙ КОМП'ЮТИНГ, КІБЕРФІЗИЧНИЙ ПРОСТІР, ПОШУК ДАНИХ, ДІАГНОСТУВАННЯ ДЕФЕКТІВ, МЕТРИКА ПОДІБНОСТІ-ВІДМІННОСТІ, КЛАС ЕКВІВАЛЕНТНОСТЕЙ, КУБИТНІ ВЕКТОРИ, МАТРИЧНІ СТРУКТУРИ ДАНИХ, ЦИФРОВІ СИСТЕМИ НА КРИСТАЛАХ

У магістерській роботі розглядаються питання, пов'язані зі створенням моделей, методів, архітектур на основі матричних структур даних для пошуку дефектів у цифрових системах.

Мета дослідження – суттєве зменшення обчислювальної складності алгоритмів пошуку даних та діагностування несправностей шляхом розробки ефективної інфраструктури матричних даних для технологічного апаратно-орієнтованого паралельного аналізу дефектів з використанням метрики подібності-відмінності.

Задачі дослідження: проаналізувати сучасні технологічні тенденції; виконати аналітичний огляд математичного апарату – моделей та методів пошуку та діагностування дефектів; розробити архітектуру секвенсора для пошуку подібності-відмінності-включення на основі матричних структур даних.

Об'єкт дослідження – цифрові системи на кристалах.

Предмет дослідження – матричні структури даних.

## ABSTRACT

The explanatory note contains: 56 pages, 25 figures, 2 tab., 22 sources according to the list of links.

MODEL, METHOD, ARCHITECTURE, CYBERPHYSICAL COMPUTING, CYBERPHYSICAL SPACE, DATA SEARCH, DEFECT DIAGNOSIS, SIMILARITY-DIFFERENCE METRICS, EQUIVALENCE CLASS, QUBIT VECTORS, MATRIX DATA STRUCTURES, DIGITAL SYSTEMS ON CRYSTALS

The master's work examines issues related to the creation of models, methods, architectures based on matrix data structures for finding defects in digital systems.

The purpose of the research is to significantly reduce the computational complexity of data retrieval and fault diagnosis algorithms by developing an effective matrix data infrastructure for technological hardware-oriented parallel analysis of defects using the similarity-difference metric.

Research tasks: to analyze modern technological trends; perform an analytical review of the mathematical apparatus – models and methods of finding and diagnosing defects; design a sequencer architecture for similarity-difference-inclusion search based on matrix data structures.

The object of research is digital systems on crystals.

The subject of research is matrix data structures.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Огляд технологій .....	11
1.2 Тенденції розвитку цифрових систем на кристалах.....	16
1.3 Висновки до розділу 1 .....	22
2 МЕТРИКИ ТА ВИМІРИ У КІБЕРФІЗИЧНОМУ ПРОСТОРИ.....	24
2.1 Інструментарій для векторного та паралельного виконання процедур .	24
2.2 Метрика та операції подібності-відмінності для пошуку дефектів при аналізі стовпців .....	28
2.4 Висновки до розділу 2 .....	31
3 МЕТОДИ ДІАГНОСТУВАННЯ ДЕФЕКТІВ .....	32
3.1 Метод різницевого діагностування.....	32
3.2 Кубітно-різницевий метод пошуку дефектів .....	34
3.3 Метод еквівалентування векторів станів матриці дефектів .....	39
3.4 Висновки до розділу 3 .....	43
4 АПАРАТНА РЕАЛІЗАЦІЯ СЕКВЕНСОРА .....	44
4.1 Секвенсор апаратного обчислення подібності-відмінності-включення об'єктів .....	44
4.2 Валідація апаратної реалізації .....	52
4.3 Верифікація програмного модуля.....	52
4.4 Висновки до розділу 4.....	54
ВИСНОВКИ.....	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	57
ДОДАТОК А Графічний матеріал до кваліфікаційної роботи (презентація)	60
ДОДАТОК Б Тези доповіді, сертифікат .....	69
Відомості кваліфікаційної роботи.....	74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ

AI – штучний інтелект (Artificial Intelligent);

CMOS – Complementary Metal-Oxide-Semiconductor;

DC – кількість з'єднань між логікою та основною пам'яттю на квадратний міліметр

DL – густина чи кількість логічних транзисторів на квадратний міліметр;

DM – число комірок пам'яті системи на квадратний міліметр;

DRAM – dynamic random access memory (динамічна пам'ять з довільним доступом);

FinFET – Fin Field-Effect Transistor;

GMT – gate-metal-tiers;

LMC – логіка (L), пам'ять (M) та зв'язок (C);

MOS-транзистор – metal-oxide-semiconductor;

MRAM – магніторезистивна оперативна пам'ять;

SoC – System-on-Chip;

SRAM – статична пам'ять з довільним доступом (static random access memory).

## ВСТУП

У магістерській роботі розглядаються питання, пов'язані зі створенням моделей, методів, архітектур на основі матричних структур даних для пошуку дефектів у цифрових системах.

Мета дослідження – суттєве зменшення обчислювальної складності алгоритмів пошуку даних та діагностування несправностей шляхом розробки ефективної інфраструктури матричних даних для технологічного апаратно-орієнтованого паралельного аналізу дефектів з використанням метрики подібності-відмінності.

Задачі дослідження: проаналізувати сучасні технологічні тенденції; виконати аналітичний огляд математичного апарату – моделей та методів пошуку та діагностування дефектів; розробити архітектуру секвенсору для пошуку подібності-відмінності-включення на основі матричних структур даних.

У роботі пропонується короткий огляд окремих новітніх програмно-апаратних рішень, орієнтованих створення ефективних засобів і систем штучного інтелекту, які дозволяють здійснювати пошук даних у кіберпросторі за прийнятний час. Розглядається універсальна метрика пошуку даних у кіберпросторі на основі використання параметрів подібності-відмінності та матричної структури в двійковій формі. Розглядаються методи аналізу матричних структур даних за метрикою подібності-відмінності для пошуку дефектів у цифрових системах. Прівнюються три методи аналізу матричних структур даних, які використовують двійковий вектор поділу матриці, а також алгоритми пошуку необхідних даних щодо аналізу рядків та/або стовпців. Перший, різницевий метод діагностування характеризується виконанням трьох логічних операцій над двійковими станами вектор-рядок матриці та орієнтований на пошук одиночних та кратних дефектів у цифрових системах та програмних додатках. Другий, кубитно-різницевий метод пошуку дефектів

характеризується використанням векторних паралельних логічних операцій на формування діагнозу багатозначного технічного стану за принципом «розділяй і поєднуй, виключаючи протиріччя». Третій метод еквівалентування векторів характеризується здатністю визначати безліч несправних станів або клас еквівалентних дефектів за заданим вектором експериментальних перевірок шляхом обчислення метричних нормованих оцінок подібності-відмінності. Показується перевага кубітно-векторного подання даних у комірках матриці, яке дає можливість у паралельному режимі виконувати логічні операції за три автоматні такти з метою отримання необхідного результату. Реалізуються методи для пошуку одиночних та кратних дефектів у цифрових програмних та апаратних системах, орієнтовані на апаратну реалізацію виконання паралельних логічних реєстрових операцій, які забезпечують суттєве підвищення швидкодії порівняно з існуючими аналогами. Вказуються обмеження запропонованих методів пошуку даних, пов'язані з необхідністю виконання процедури синтезу матричних структур даних, що має квадратичну обчислювальну складність. Пропонується апаратна реалізація секвенсору для пошуку подібності-відмінності-включення, який характеризується отриманням більш точної структурованої оцінки взаємодії двох об'єктів.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Наводиться стан сучасного комп'ютингу на основі огляду літературних джерел та технологічних трендів Gartner Research Group. Пропонується короткий огляд окремих новітніх програмно-апаратних рішень, орієнтованих створення ефективних засобів і систем штучного інтелекту, які дозволяють здійснювати пошук даних у кіберпросторі за прийнятний час.

### 1.1 Огляд технологій

Історично склалося, що апаратні та програмні засоби комп'ютера на кожній стадії його розвитку становлять між собою гармонійний альянс. Це означає, що новим чіпам повинні відповідати нові програмні системи, так само як і під нові алгоритми та технології слід розробляти більш досконалі кремнієві кристали та структури.

Відомо, що провідна компанія планети Apple використовує стратегію вмирання старого заліза шляхом примусової невідтримки новими програмними системами через певний час. Так виключається мезальянс між старим залізом та новими програмним забезпеченням. Аналогічна ситуація має місце щодо нового заліза та старих додатків.

Зростання обчислювального потенціалу нових чіпів надає вченим можливість проектувати досконаліші моделі, методи та алгоритми вирішення актуальних ринкових та наукових задач. Тому кожен фахівець із програмування, який бажає бути в тренді, повинен мати уявлення про прориви нового покоління чіпів, комп'ютерів, мереж, дату центрів та хмарних сервісів.

Найвідомішим принципом у мікро-нано-електроніці протягом понад 55 років є закон Мура (рис. 1.1): "Кожні 2 роки кількість транзисторів на фіксованій площі кремнію подвоюється" [1].

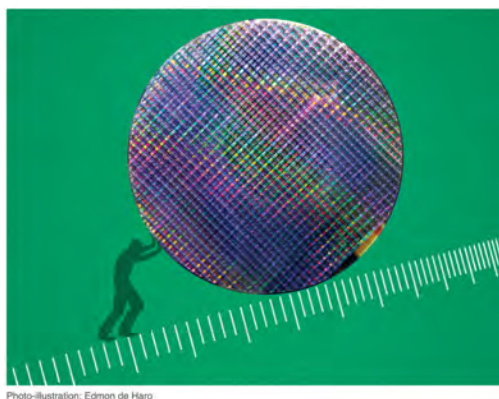


Photo-illustration: Edmon de Haro

Рисунок 1.1 – Закон Мура

Сьогодні процесор iPhone зроблений з використанням 7-нанометрової технології, коли на одному квадратному міліметрі кремнію розміщується близько 100 мільйонів транзисторів. Процесори, спроектовані на 5-нм вузлі (node = gate length), вже у виробництві. До середини 1990-х років вузли логічної технології були синонімічні із довжиною затвора транзисторів CMOS. Фактична довжина gate length деякий час скорочувалася швидше, та перестала зменшуватися. Лідери галузі заявляють, що протягом десятиліття ринок отримає 1-нм ступінь дозволу транзисторів та міжз'єднань на кристалі кремнію, що відповідає розмірності п'яти атомів кремнію. З 1971 року, коли було випущено мікропроцесор Intel 4004, лінійні розміри MOS-транзистора скоротилися приблизно 1000 разів, які кількість одному кристалі збільшилася приблизно 15 мільйонів раз (рис. 1.2).

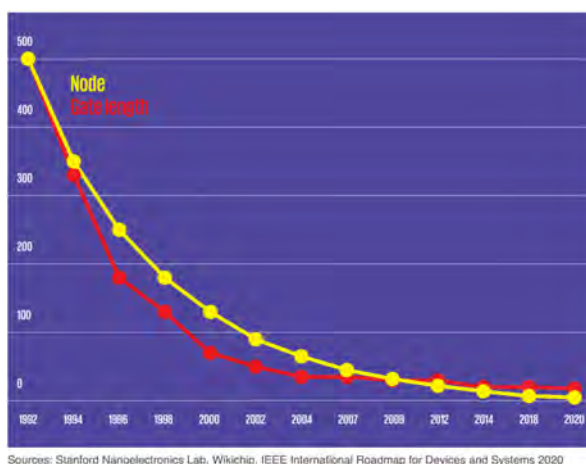


Рисунок 1.2 – Лінійні розміри MOS-транзистора

Однак найпередовіша ультрафіолетова літографія має межу на світі з довжиною хвилі 13,5 нм. Тому виробники чіпів звертаються до монолітної 3D-інтеграції, додаючи шари в кристал, щоб зберегти збільшення щільності кремнієвої CMOS. Метод-метрика GMT (gate-metal-tiers) враховує розмірність контактного кроку (pitch) для gate та metal, а також кількість шарів на кристалі (рис. 1.3). Ця метрика показує, що сьогодні 5-нм чіпи мають контактний крок затвора 48 нм, металевий крок 36 нм і лише один рівень tier, що формує карту G48M36T1, яка передає набагато кориснішу інформацію, ніж 5-нм вузол.

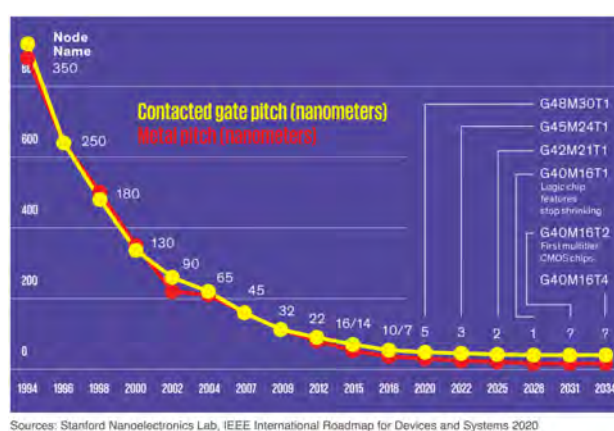


Рисунок 1.3 – Розмірність контактного кроку (pitch) для gate та metal

Проблема розсіювання енергії виникла на початку 2000-х років та була вирішена запровадженням нижчих напруг, підвищенням швидкодії та енергоефективності CMOS-пристроїв без зменшення довжини затвора. Далі проблеми витоку струму зажадали структурних змін у транзисторі CMOS. У 2011 році компанія Intel перейшла на FinFET транзистори з вузлом 22 нм, де пристрої мали довжину 26 нм затвора. Вчені Массачусетського технологічного інституту пішли шляхом створення тривимірних чіпів, де рівні транзисторів перемежуються з вуглецевими нанотрубками, що дає можливість простіше створювати чіпи при відносно низьких температурах, ніж шари кремнієвих кристалів, що вимагають високих температур для виготовлення.

Світова еліта вчених у галузі чіп-наноелектроніки (Chenming Hu, Tsu-Jae King Liu, Jeffrey Bokor, Sayeef Salahuddin, H.-S. Philip Wong, Subhasish Mitra, James D. Plummer, Kerem Akarvardar, Dimitri Antonidis) усвідомлює сьогодні, що їхня наука стає менш привабливою для кращих студентів. Логіка: якщо сфера діяльності, в якій успіхи студента мало ймовірні навіть через 10 років, навіщо витратити гроші та час на навчання від чотирьох до шести років. Висновок – потрібні інноваційні технологічні та привабливі рішення для подальшого розвитку комп'ютерних технологій, інваріантні на час їх імплементації на ринку.

Вчені зробили раціональні та зрозумілі висновки. Комп'ютер – це логіка (L), пам'ять (M) та зв'язок (C) між ними. Тому вони запропонували нову LMC метрику параметрів у вигляді густини (Density) розміщення на чіпі (чіплеті) кожного зі згаданих компонентів, назвавши їх DL, DM та DC. Будь-які поліпшення параметрів DL, DM та DC суттєво впливають на продуктивність та енергоефективність обчислювальних систем, що особливо важливо в сучасну епоху обчислень, орієнтованих на обробку великих даних.

Вчені показали кореляцію між зростанням логіки, пам'яті та зв'язків, що свідчить про збалансоване зростання DL, DM та DC протягом останніх десятиліть. Така рівновага існує в комп'ютерних архітектурах різного ступеня складності – від мобільних та настільних процесорів до найшвидших суперкомп'ютерів у світі. У метриці LMC параметр DL – це густина чи кількість логічних транзисторів на квадратний міліметр (рис. 1.4); DM – число комірок пам'яті системи на квадратний міліметр (рис. 1.5); DC – кількість з'єднань між логікою та основною пам'яттю на квадратний міліметр (рис. 1.6). Якщо є кілька рівнів у чіпі, то враховується весь об'єм для квадратного міліметра, що розглядається.

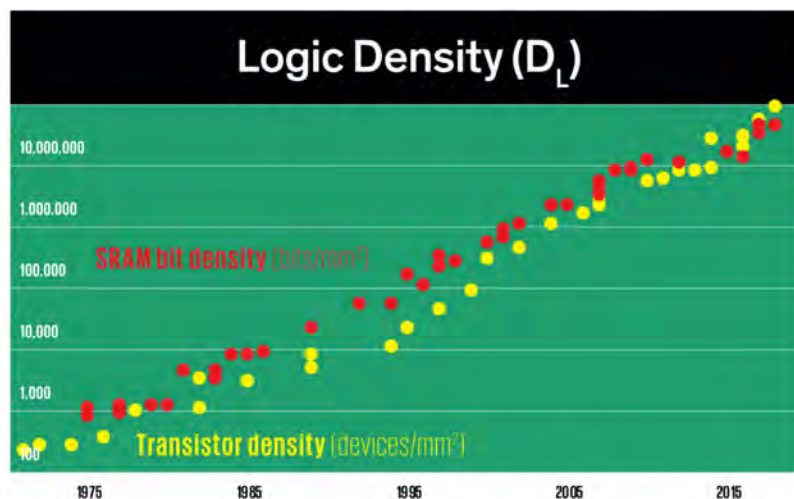


Рисунок 1.4 – Щільність логіки DL

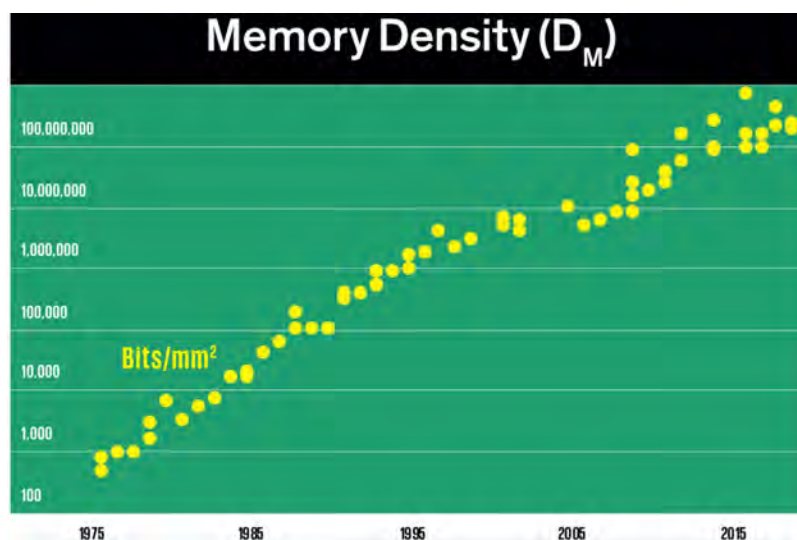


Рисунок 1.5 – Щільність пам'яті DM

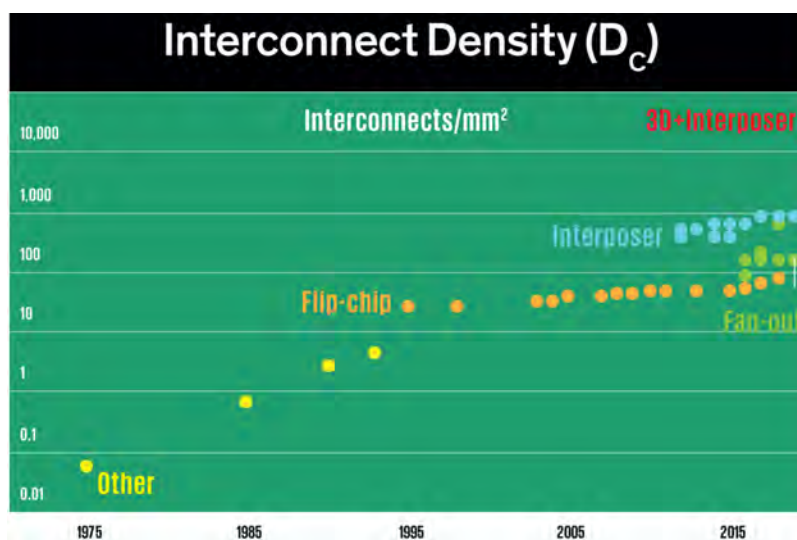


Рисунок 1.6 – Щільність з'єднань DC

Найбільш щільною частиною логічної мікросхеми є кеш пам'ять SRAM процесора, де дані зберігаються в осередках шести транзисторів (NAND logic – 4 transistors) для швидкого повторного доступу. Найбільше значення для DL на сьогоднішній день має 135-мегабітний 5-нм масив SRAM, який упакований із роздільною здатністю 286 мільйонів транзисторів на квадратний міліметр. Але блоки логіки є складнішими, менш однорідними і менш щільними, ніж вбудована у них SRAM. Різні типи процесорів: графічні, нейронні, цифрові фільтри мають різні співвідношення логіки і SRAM. Тому творці LMC вирішили не вказувати конкретний спосіб вимірювання DL, залишаючи його відкритим для обговорення галузі. Вимір DM для основної пам'яті DRAM трохи простіше. Вона недорога, має високу надійність і швидкодію читання-запису. Комірка DRAM складається з одного транзистора, який контролює доступ до конденсатора, що зберігає біт, як заряд, який повинен періодично оновлюватися. Найбільше значення DM має Samsung DRAM із щільністю 200 мільйонів осередків на квадратний міліметр. З'являються альтернативні технології пам'яті: магніторезистивна оперативна пам'ять (MRAM), сегнетоелектрична, резистивна та меристорна «аналогова» пам'ять із фазовим переходом. Деякі з них є пам'яттю, вбудованою в сам процесор, а деякі виконуються у вигляді автономних мікросхем. Забезпечення зв'язку між основною пам'яттю та логікою є вузьким місцем у сучасних обчислювальних системах.

## 1.2 Тенденції розвитку цифрових систем на кристалах

В останнє десятиліття однокристальні структури (SoC) почали поступатися мікросхемам, тривимірно пов'язаних один з одним на кремнієвих підкладках (2.5-D системи), які мають 12000 міжз'єднань на квадратний міліметр. Найкраща метрика параметрів системи, де DL, DM та DC відповідно дорівнюють [260M, 200M, 12K]. "Вибір того, що узгоджено, навіть якщо воно недосконале, корисніше, ніж нинішній брендинг вузла", Майкл Мейберрі,

Intel. Інші показники якості чіпа: продуктивність, потужність, вартість, ціна за біт пам'яті, інтегрованість з іншими компонентами. Інша точка зору: інтеграція CMOS-логіки з енергонезалежною пам'яттю та міліметрове радіо є більш важливою для майбутнього галузі, ніж масштабування. Резюме: масштабування метрики LMC і GMT транзисторів не таке важливе для залучення технічних талантів при реалізації нано-майбутнього, але без синхронізації метрики галузь стає менш ефективною. Є 10 років, після чого кремнієва CMOS перестане скорочуватися.

IBM вважає, що головним масштабуючим елементом для мікросхем, поза FinFET, буде транзистор-нанопластина (nanosheet), як останній крок у законі Мура [2].

Нанолист йде заміну архітектури FinFET і уможливить перехід від вузлів 7 – 5 нм до вузла 3 нм (рис. 1.7). В архітектурі польових транзисторів на нанолистах струм протікає через безліч кремнієвих пакетів, оточені затвором транзистора. Така конструкція значно зменшує силу струму у вимкненому стані та використовує більший струм при керуванні включеним транзистором. У майбутньому побудова обчислювальних систем із гетерогенною архітектурою та спеціалізованою пам'яттю стане ключовим фактором при реалізації AI-пристроїв.

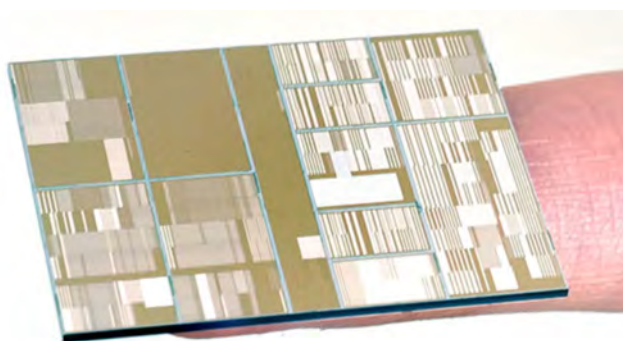


Рисунок 1.7 – Нанолист замість FinFET архітектури

1. Замість материнських плат [3] компанія SEA-Leti пропонує чіплет (рис. 1.8), який інтегрує цифрові системи на кристалах різних компаній з функціональностей та електроживлення. Система чіпів може пропускати 3 терабайти в секунду на одному квадратному міліметрі кремнію (150мл/1кв.мм потужність розсіювання) з затримкою 0,6 наносекунди на міліметр. Чіп SEA-Leti укладає шість 16-ядерних чіпсетів поверх тонкої пластини (інтерпозера) кремнію. Інтерпозер містить як схеми регулювання напруги, і функціональну мережу, яка пов'язує різні компоненти з вбудованою пам'яттю до 2 терабайтів. Активні пристрої, що вбудовуються, є кращим шляхом інтегрування в системи розрізаних технологій і мікросхем від різних постачальників.



Рисунок 1.8 – Система чіплетів на силіконовій підкладці

2. Ізраїльський стартап Nailo [4] зібрав 60 мільйонів доларів для виробництва чіпа Nailo-8. Він призначений для глибокого навчання на автомобілях, роботах та інших "передових" машинах. Інноваційні мікросхеми призначені для зменшення витрат, розмірів та енергоспоживання при використанні П для обробки високої роздільної здатності даних від датчиків, камер HD. Nailo ідентифікує свій проект структурно визначеною архітектурою для потоку великих даних. Навчання називається "глибоким", тому що його нейронні мережі складаються з довгого стеку нейронних шарів між входом та виходом. Програмні засоби Nailo використовують навчену нейронну мережу для перевірки можливостей пам'яті, обчислень та потоку даних, навантажених на кожен рівень (рис. 1.9). Додаток зіставляє інформацію з ресурсами

процесора для формування структури, де кожен рівень є суміжним із своїми сусідами. В результаті виходить мікросхема, яка може виконувати 26 трильйонів операцій за секунду і досягати 672 кадри за секунду за зразком класифікації зображень ResNet-50 з ефективністю 3,1 трильйона операцій за секунду на ват. Дається картина Hailo-8 у порівнянні з іншими прискорювачами ШІ у тестах MLPerf. Чіпи призначені для використання в периферійних системах, включаючи частково автономні транспортні засоби, інтелектуальні камери, смартфони, дрони та гаджети AR/VR, які потребують складного глибокого навчання для точного виконання заданих функцій. яка може виконувати 26 трильйонів операцій на секунду і досягати 672 кадри на секунду за стандартом класифікації зображень ResNet-50 з ефективністю 3,1 трильйона операцій на секунду на ват.

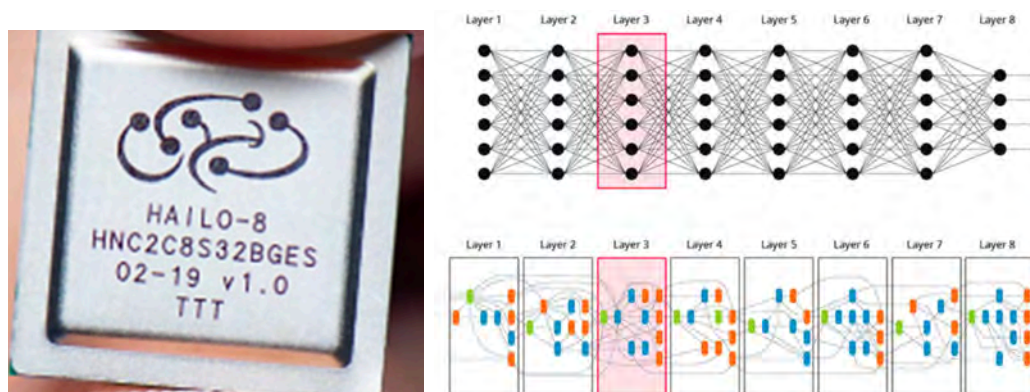


Рисунок 1.9 – Чіп Hailo та його структура для Machine Learning

3. Комп'ютер CS-1 [5] мав конструкцію, де близько трьох чвертей кожного шасі займали системи охолодження. У 2020 році відбулася справжня революція: з'явився надзвичайно потужний комп'ютер, який майже повністю складається з одного чіпа (рис. 1.10), розмір якого становить понад 46 255 квадратних міліметрів, що більш ніж у 50 разів перевищує площу будь-якого іншого процесорного чіпа. Маючи 1,2 трлн транзисторів, 400 000 процесорних ядер, 18 гігабайт SRAM і міжкомпонентні з'єднання, здатні переміщати 100 мільйонів мільярдів біт на секунду, Wafer Scale Engine (WSE) Cerebras не

піддається простому порівнянню з іншими системами. За даними компанії, 10-стійковий кластер TPU2 компанії Google AI споживає в п'ять разів більше енергії та займає в 30 разів більше місця, забезпечуючи лише одну третину продуктивності від комп'ютера WSE Cerebras.

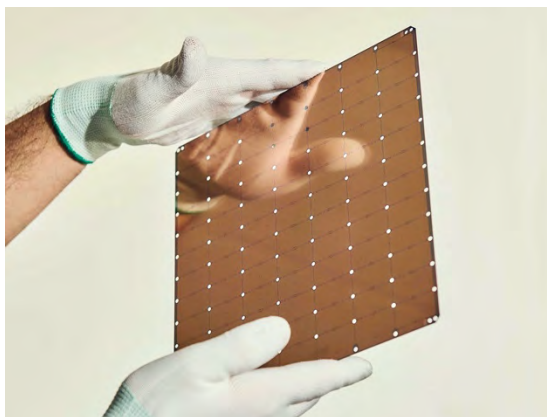


Рисунок 1.10 – Нейро-комп'ютер Wafer Scale Engine Cerebras

4. Наступний найпримітивніший пристрій дає основу для створення як потужних дата центрів і суперкомп'ютерів, так і кінцевих гаджетів і сенсорних пристроїв. Польовий (field-effect transistor) транзистор [6] складається з чотирьох основних частин: витoku (source), стоку (drain), каналу (conductive channel), який з'єднує їх, та затвора (gate) для керування потоком струму по каналу (рис 1.11).

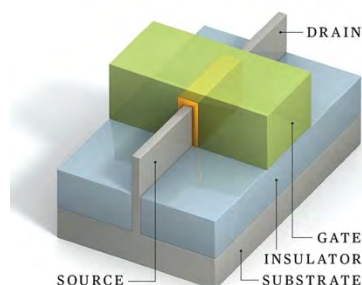


Рисунок 1.11 – Справжнє та майбутнє мікроелектроніки – FinFET або Fin Field-Effect Transistor

Оскільки всі компоненти стали меншими, то було помічено, що поведінка транзисторів суттєво змінюється у бік порушення працездатності при тривалому їх використанні. Chenming Hu – батько FinFETs – вважав фундаментальну проблему цілком ясною для себе – зробити канал дуже тонким, щоб електрони не могли проходити крізь затвор. Для цього використовували стоншення оксидного шару затвора з метою кращого контролю над каналом та зменшення струму витoku. Його дослідження показали, що підхід стоншення близький до межі, а саме: зробити шар оксиду досить тонким, щоб електрони могли перестрибнути через нього в кремнієву підкладку, створюючи ще одне джерело витoku. Натомість він запропонував дві інші ідеї. Перша – додати шар ізоляції, прихований у кремнії під транзистором, для скрути електронам пробиратися крізь затвор зарядів. Цю конструкцію стали називати повністю збідненим кремнієм на ізоляторі або FDSOI (fully depleted silicon-on-insulator). Друга пов'язана з наданням затвору більшого контролю над потоком заряду шляхом розширення тонкого каналу вертикально над підкладкою, подібного до акулячого плавця (shark's fin), щоб затвор міг обрамляти або огортати (wrap) канал з трьох сторін, а не просто сидіти зверху. Ця структура називається FinFET (Fin Field-Effect Transistor), яка має додаткову перевагу, коли використання простору вертикалі усуває деякі затори на площині 2D, відкриваючи еру тривимірних транзисторів. Особливості FinFET: кожен транзистор має витік, стік, канал і затвор для керування струмом каналом. Підняття каналу над поверхнею чіпа дозволяє затвору обертати його із трьох сторін, створюючи більший контроль.

Новий надшвидкий пристрій (рис. 1.12) машинного зору [7] може обробляти зображення в тисячі разів швидше за існуючі аналоги за допомогою датчика зображення, що використовує штучну нейронну мережу. Технологія машинного зору часто відчуває затримки сканування пікселів рядок за рядком, перетворення відеокадрів на цифрові сигнали та передачі даних на комп'ютери для аналізу. Вчені з TU Wien прискорили машинний зір, прибравши посередника. Вони створили датчик зображення – штучну нейронну мережу,

яка може одночасно отримувати та аналізувати дані. Потім вчені створили датчик – нейронну мережу, що навчається, на основі зв'язків між фотодіодами для класифікації зображень, включаючи літери. Датчик зображення не споживає електроенергію під час роботи. Самі виявлені фотони забезпечують енергію до роботи нейромережі. Технологія машинного зору зазвичай здатна обробляти до 100-1000 кадрів за секунду. Для порівняння: TU Wien система працює з 20 мільйонами кадрів за секунду. Швидкодія системи обмежена лише швидкістю електронів у ланцюгах. Стратегія може працювати на рівні пікосекунд, або трильйонних часток секунди, що на три-чотири порядки швидше за існуючі аналоги. Датчик орієнтований на пошук та аналіз даних у візуалізації динаміки рідини, процесів горіння або механічних поломок. Машинний зір в автономному керуванні вимагає інших ефективних рішень.

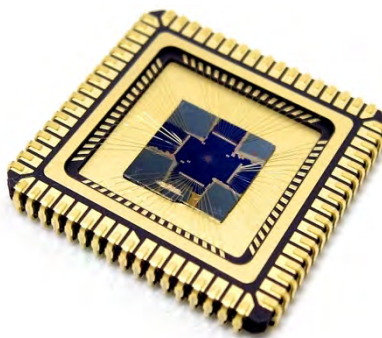


Рисунок 1.12 – Нейропроцесор для аналізу візуальних потоків даних

### 1.3 Висновки до розділу 1

Проналізовано стан сучасного комп'ютингу на основі огляду літературних джерел та прогнозів американської дослідницької та консалтингової компанії Gartner Research Group.

Таким чином, навіть окремі найкращі рішення в галузі комп'ютингу, аналізу та пошуку даних свідчать про існування стійкого тренду на інтелектуалізацію «заліза» шляхом імплементації апаратних рішень в

алгоритми штучного інтелекту та машинного навчання, що створюють засоби для високопродуктивних обчислень, необхідних сьогодні у кіберфізичному просторі для аналізу великих даних. Загальновідомо, що це моделі, методи, алгоритми і застосування, пов'язані з штучним інтелектом, прагнуть у розвитку від ймовірнісних характеристик процесів до детермінізму кінцевих автоматів, закладених спочатку в сучасний комп'ютинг.

Відомо, що ідеальним описом деякого процесу чи явища завжди була, є і буде строго детермінована таблиця істинності від змінних  $n$ , як певну межу знань про об'єкт дослідження [8]. Тому шляхи вирішення актуальних завдань із розпізнавання патернів, образів, станів формуватимуться у зустрічних напрямках: 1) від ймовірнісного незнання процесу до його автоматного детермінізму через час навчання; 2) від комп'ютерного детермінізму як ядра деякого знання до багатопараметричної деталізації аналізованого процесу або явища. Другий шлях передбачає розробку ефективної метрики подібності-відмінності, структур даних та алгоритмів для аналізу та навчання спочатку детермінованих систем. як ядра деякого знання, до багатопараметричної деталізації аналізованого процесу чи явища.

Мета дослідження – суттєве зменшення обчислювальної складності алгоритмів пошуку даних та діагностування несправностей шляхом розробки інфраструктури для апаратно-орієнтованого паралельного аналізу дефектів на основі метрики подібності-відмінності.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- проаналізувати сучасні технологічні тенденції;
- виконати аналітичний огляд математичного апарату – моделей та методів пошуку та діагностування дефектів;
- розробити архітектуру секвенсору для пошуку подібності-відмінності-включення на основі матричних структур даних.

## 2 МЕТРИКИ ТА ВИМІРИ У КІБЕРФІЗИЧНОМУ ПРОСТОРИ

Розглядаються метрики на основі подібності-відмінності об'єктів, процесів або явищ як базис для розв'язання задач, пов'язаних з вимірами в кіберфізичному просторі.

### 2.1 Інструментарій для векторного та паралельного виконання процедур

Метрика подібності-відмінності об'єктів процесів або явищ (рис. 2.1), представлена рівнянням  $S \oplus D = a \cup b = U = 1$ , є базисом для вирішення найпоширеніших на технологічному ринку завдань, пов'язаних з вимірами в кіберфізичному просторі та інтернеті [9].

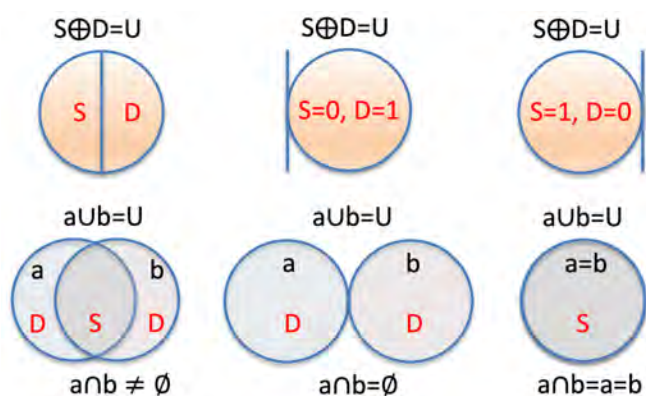


Рисунок 2.1 – Метрика відношень між подібністю-відмінністю двох об'єктів

Такими науково-практичними задачами (рис. 2.2) є: 1) розпізнавання образів та патернів; 2) машинне навчання; 3) ухвалення рішень; 4) управління процесами та об'єктами; 5) аналітика великих даних; 6) цифровізація процесів та явищ; 7) тестування та діагностика систем; 8) створення та використання регулярних баз даних.

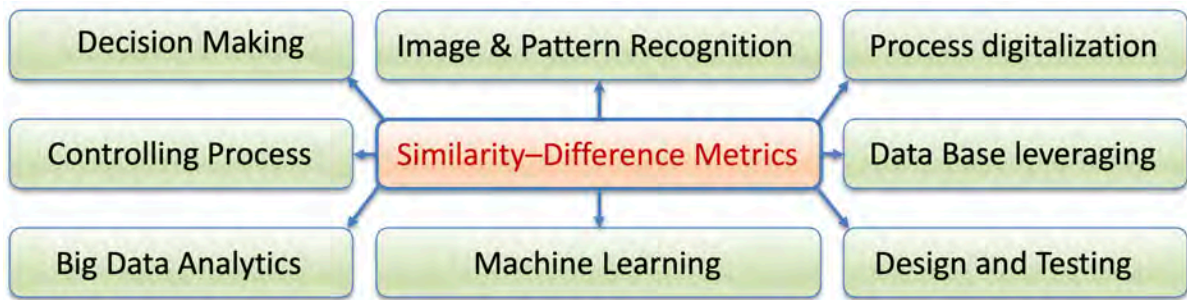


Рисунок 2.2 – Задачі метрики відношень між подібністю-відмінністю двох об'єктів

Модель відношень між процесами та/або явищами оптимально представляється у вигляді двійкової (бінарної) матриці  $M=[M_{ij}]$ , яка формує в загальному випадку декартовий добуток двох множин, наприклад, сукупність тестів на множині функціональностей:  $\langle T \times F \rangle$ .

Подібність – відношення кількості однакових компонентів процесу або явища до їх загального числа у метриці заданих істотних параметрів. Відмінність – відношення кількості відмінних один від одного компонентів процесу або явища до їх загального числа в метриці заданих істотних параметрів. Подібність і відмінність взаємно доповнюють одне одного як оцінки та відношення між процесами та явищами.

Для визначення скалярної оцінки подібності між усіма парами компонентів необхідно з матриці  $M$  побудувати квадратичну матрицю їхньої подібності ( $T \times T$  or  $F \times F$ ) шляхом використання формули, що визначає відношення подібності двох компонентів:

$$M_{ik} = S(M_i, M_k) = \frac{\sum_{j=1}^n (M_{ij} \wedge M_{kj})}{\sum_{j=1}^n (M_{ij} \vee M_{kj})}$$

Подібність визначається непустим перетином вектор-рядка матриці  $M$  з диз'юнкцією всіх інших рядків. Формула для обчислення скалярної оцінки подібності між компонентами зводиться до визначення подібності між

поточним рядком  $M_k$  та диз'юнкцією-об'єднанням всіх інших рядків  $\tilde{M}_k = \vee M_{ij}$ , знайдених у матриці  $M$  (один з усіма, але у форматі вектора  $M_k$ ):

$$S(M_k, \tilde{M}_k) = \frac{\sum_{j=1}^n [M_{kj} \wedge (\bigvee_{i=1, m}^{i \neq k} M_{ij})]}{\sum_{j=1}^n M_{kj}}$$

Результатом виконання даної формули є скалярна оцінка, яка формує нормоване відношення кількості одиничних координат, загальних для аналізованого компонента і вектора логічного об'єднання всіх інших компонентів, до одиничних координат аналізованого рядка. Сукупність таких скалярних оцінок всім рядків матриці  $M$ :  $S(M_k, \tilde{M}_k)$ ,  $k = \overline{1, m}$  дає можливість ранжувати їх за рівнем зменшення подібності – відмінності. Природно, що рядки з максимальними оцінками подібності слід виключати з множини валідних компонентів, перетин яких між собою повинен в ідеалі дорівнювати порожній множині.

Доповнення: якщо порахувати відмінність для компонента  $D = 1 - S(M_k, \tilde{M}_k)$ , а потім всіх їх підсумовувати, то вийде оцінка сукупностей частин компонентів, що не перетинаються, які формують інтегральну функціональність матриці. Розподіл даної оцінки на довжину матриці визначатиме процент корисної функціональності:

$$D(M_k, \tilde{M}_k) = \frac{\sum_{j=1}^n [M_{kj} \wedge (\bigvee_{i=1, m}^{i \neq k} M_{ij})] \oplus M_{kj}}{\sum_{j=1}^n M_{kj}}$$

Після виключення невалідних компонентів необхідно виконувати процедуру верифікації шляхом визначення їх спільної загальної подібності, сенс якої полягає у підрахунку одиничних координат після одночасного перетину всіх векторів  $M_i \in M$ , віднесених до суми одиничних координат, які

отримані після об'єднання векторів, що в сукупності дає нуль як позитивний результат:

$$S(M) = \frac{\sum_{j=1}^n (\wedge_{i=1}^m M_{ij})}{\sum_{j=1}^n (\vee_{i=1}^m M_{ij})}.$$

Ця формула визначає рівень спільності, якщо значення отриманого результату відмінно від нуля. У цьому випадку необхідно виключати з багатьох векторів, які мають високий рівень рейтингу за метрикою подібності з іншими рядками.

Для масштабованого сприйняття оцінок подібності-відмінності необхідно також виводити як супровідний результат базу-метрику, яка являє собою в даному випадку кількість одиничних координат аналізованого вектора. Аналогічно слід надходити і при верифікації компонентів шляхом їхнього спільного перетину, де потрібно виводити інформацію про сукупну кількість одиничних координат, отриманих в результаті об'єднання всіх векторів, що фігурують у знаменнику. Тоді вихідні дані матимуть формат «схожість – число одиниць вектора»:  $S(M_k, \tilde{M}_k), (\sum_{j=1}^n M_{kj})$ .

Інтегральний рівень подібності в матриці  $C(M)$  є більш структурованою скалярною оцінкою порівняно з  $S(M)$ , яка визначається шляхом підсумовування 1-координат після об'єднання подібності всіх векторів щодо диз'юнкції інших, поділеної на суму 1-координат після об'єднання всіх рядків матриці  $M$ :

$$C(M) = \frac{\sum_{j=1}^n \{ \vee_{k=1}^m [M_{kj} \wedge (\vee_{i=1, m}^{i \neq k} M_{ij})] \}}{\sum_{j=1}^n (\vee_{k=1}^m M_{kj})}.$$

Якщо результат  $C(M) = 0$ , то кожен вектор не має подібності до інших, що означає коректне розпізнавання компонентів-рядків у матриці  $M$ . В іншому випадку необхідно видаляти ті рядки, які мають максимальні оцінки



U	0	1	X	∅	...	v	10011100
0	0	X	X	0	...	10101111	0
1	X	1	X	1	→	0100010111	
X	X	X	X	X	...	1111111111	
∅	0	1	X	∅	...	0010011100	

a	0	1	X	∅	...	a	10011100
$\bar{a}$	1	0	∅	X	→	$\bar{a}$	01100011

Δ	0	1	X	∅	...	⊕	10011100
0	∅	X	1	0	...	1000110110	
1	X	∅	0	1	→	0111001001	
X	1	0	∅	X	...	1101100011	
∅	0	1	X	∅	...	0010011100	

Слід зазначити, що виконання логічних операцій над векторами-кубітами є більш технологічним для сприйняття людиною та комп'ютером: перетину-кон'юнкції, об'єднання-диз'юнкції, доповнення-заперечення, симетричної різниці – виключаюче АБО. По суті, дані логічні операції становлять сутність подібності-відмінності процесів або явищ, що фігурують у метричному рівнянні пошуку даних [9]:  $S \oplus D = a \cup b = \overline{a \cap b}$ . Поділяй та об'єднуй, виключаючи протиріччя – основна ідея пошуку.

У випадку діагностування згадана ідея може бути представлена розподілом множини векторів матриці на M два підмножини за параметром  $R = \{0,1\}$ , як показано на рис. 2.4.

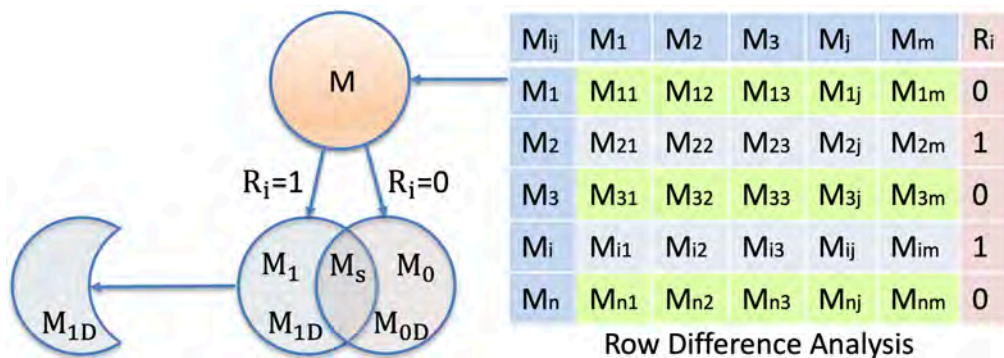


Рисунок 2.4 – Аналіз відмінності за рядками матриці

Отримані підмножини векторів множини  $M$  у загальному випадку взаємодіють між собою за подібністю, що визначається відповідно до формули:  $M_s = M_1 \cap M_0$ .

Враховуючи той факт, що множина  $M$  і похідні підмножини створюються за допомогою елементів, що утворюють вектори, актуальними є процедури, що формують подібності та відмінності між підмножинами  $M_1, M_0$ :

$$M_{1D} = M_1 \setminus (M_1 \text{ or } M_s);$$

$$M_{1D} = M_1 \setminus M_1 \cap M_0;$$

$$M_{1D} = M_1 \cap \overline{M_1 \cap M_0};$$

$$M_{1D} = M_1 \wedge \overline{M_1 \wedge M_0}.$$

Модифікація даних формул у бік більш точного розв'язку задач діагностування на основі додаткового аналізу схожості окремих векторів множини  $M_1$  призводить до очевидного результату:

$$M_{1D} = \cap M_1 \setminus (M_1 \text{ or } M_s);$$

$$M_{1D} = \cap M_1 \setminus M_1 \cap M_0;$$

$$M_{1D} = \cap M_1 \cap \overline{M_1 \cap M_0};$$

$$M_{1D} = \wedge M_1 \wedge \overline{M_1 \wedge M_0}.$$

Тут додається операція пошуку загальної частини або подібності між усіма векторами, які відносяться до підмножини  $M_1$ . Стосовно класу константних несправностей, подібність асоціюється до пошуку загальних одиночних дефектів, які присутні у всіх векторах  $M_1$  і відсутні у всіх векторах множини  $M_0$ . Щодо пошуку кратних дефектів, то остання система рівнянь трансформується в сукупність виразів:

$$M_{1D} = \cup M_1 \setminus (M_1 \text{ or } M_s);$$

$$M_{1D} = \cup M_1 \setminus M_1 \cap M_0;$$

$$M_{1D} = \cup M_1 \cap \overline{M_1} \cap \overline{M_0};$$

$$M_{1D} = \vee M_1 \wedge \overline{M_1} \wedge \overline{M_0}.$$

Подані вирази орієнтовані підвищення глибини діагностування, що означає зменшення потужності підмножини  $M_{1D}$ .

## 2.4 Висновки до розділу 2

Проаналізовано інструментарій для векторного та паралельного виконання процедур та описано метрики на основі подібності-відмінності об'єктів, процесів або явищ як базис для розв'язання задач, пов'язаних з вимірами в кіберфізичному просторі.

### 3 МЕТОДИ ДІАГНОСТУВАННЯ ДЕФЕКТІВ

Розглядаються та аналізуються методи діагностування дефектів: різницевий метод пошуку, заснований на векторному поданні множини несправностей, що перевіряються на тестових наборах; матричний метод діагностування несправних станів цифрового виробу, що використовує як діагностичну інформацію двійкову матрицю; кубітно-різницевий метод пошуку кратних дефектів на основі обчислення теоретико-множинної різниці двох векторів-рядків матриці; метод аналізу стовпців-векторів.

#### 3.1 Метод різницевого діагностування

Для діагностування дефектів можна використовувати різницевий метод пошуку, заснований на векторному поданні [10–12] множини несправностей  $F$ , що перевіряються на тестових наборах  $T$ :

$$F_m^1 = \bigcup_{\forall R_i=1} T_i = \bigvee_{\forall R_i=1} T_i;$$

$$F_s^1 = \bigcap_{\forall R_i=1} T_i = \bigwedge_{\forall R_i=1} T_i;$$

$$F^0 = \bigcup_{\forall R_i=0} T_i = \bigvee_{\forall R_i=0} T_i;$$

$$D_m = F_m^1 \cap \overline{F^0} = \bigcup_{\forall R_i=1} T_i \cap \overline{\bigcup_{\forall R_i=0} T_i} = F_m^1 \wedge \overline{F^0} = \bigvee_{\forall R_i=1} T_i \wedge \overline{\bigvee_{\forall R_i=0} T_i};$$

$$D_s = F_s^1 \cap \overline{F^0} = \bigcap_{\forall R_i=1} T_i \cap \overline{\bigcup_{\forall R_i=0} T_i} = F_s^1 \wedge \overline{F^0} = \bigwedge_{\forall R_i=1} T_i \wedge \overline{\bigvee_{\forall R_i=0} T_i}.$$

Тут фігурують два рівняння, які визначають сукупність одиночних  $D_s$  чи кратних  $D_m$  дефектів у цифровій системі. При цьому в початковій стадії діагностування слід скористатися гіпотезою про існування кратного дефекту, що є більш імовірною подією в процесі експлуатації виробу. Якщо такий алгоритм дає порожню множину дефектів, необхідно скористатися другим рівнянням для пошуку кратних несправностей. У процесі виконання діагностичного експерименту використовуються дві групи векторів, які класифікуються шляхом їх приналежності до одиничної або нульової множини, що формується на основі  $\{1,0\}$ -реакції R цифрового пристрою на тестові дії.

Далі розглядається матричний метод діагностування несправних станів цифрового виробу, що використовує як діагностичну інформацію двійкову матрицю «тест–несправність»  $M=\langle T,F\rangle$ , яка має наступний вигляд:

$M=\langle T,F\rangle$	F1	F2	F3	F4	F5	F6	F7	F8	R
T1	1		1		1				1
T2		1		1		1			0
T3	1					1			0
T4			1		1		1		1
T5	1				1				0
T6		1						1	0
T7									0
T8			1		1				1
$F_s^1 = \bigwedge_{\forall R_i=1} T_i$			1		1				
$F^0 = \bigvee_{\forall R_i=0} T_i$	1	1		1	1	1		1	
$D_s = F^1 \wedge \overline{F^0}$			1						

Отриманий результат  $D_s=F3$  є одиночною несправністю в цифровій системі, яка є у всіх трьох векторах підмножини  $M(R=1)$  і відсутня у всіх векторах підмножини  $M(R=0)$ . Що стосується умови наявності в системі кратних несправностей, то наступна таблиця ілюструє процедуру пошуку, яка оперує об'єднаннями векторів у рамках розгляду підмножин:  $M(R=1)$  і  $M(R=0)$  з наступним відніманням другого з першого:

M=<T,F>	F1	F2	F3	F4	F5	F6	F7	F8	R
T1	1		1		1				1
T2		1		1		1			0
T3	1					1			0
T4			1		1		1		1
T5	1				1				0
T6		1						1	0
T7									0
T8			1		1				1
$F_m^1 = \bigvee_{\forall R_i=1} T_i$	1		1		1		1		
$F^0 = \bigvee_{\forall R_i=0} T_i$	1	1		1	1	1		1	
$D_m = F^1 \wedge \overline{F^0}$			1				1		

Матричний різницевий метод діагностування має особливість, яка пов'язана із двійковим алфавітом опису координат матриці. Вважаючи це певним недоліком масштабованості методу, пропонується багатозначне уявлення координат, яке може бути трансформовано у вектори, розмірністю, що дорівнює значення пропонованого алфавіту. Таким чином, матриця  $M=[M_{ij}]$  формує добуток трьох множин:  $\langle T \times F \times A \rangle$ , де  $A$  – є кінцевим алфавітом опису станів цифрового пристрою.

### 3.2 Кубітно-різницевий метод пошуку дефектів

Кубіт тут визначається як двійковий вектор, розмірності  $k$ , що одночасно ідентифікує кінцеве число станів, як суперпозицію їх унітарних кодів. Кубіт логічної функції є вектор, що формує  $k=2^n$  станів виходу, де  $n$  – число змінних [10,13]. Кубіт несправностей є вектор, що формує  $k$  несправних кодів станів окремої лінії, компонента або системи, що мають здатність суперпозиції завдяки властивості унітарності їх кодів. Параметр результатів тестування  $R=(R_1, R_2, \dots, R_i, \dots, R_p)$ , заданий в алфавіті  $\{0,1\}$ , рівний довжині тесту  $p$  (де  $T=(T_1, T_2, \dots, T_i, \dots, T_p)$ ), є вектор-кубіт значень функції несправностей, що формується в процесі діагностичного експерименту. Тому будь-який рядок

матриці дефектів  $M=(M_1, M_2, \dots, M_i, \dots, M_p)$  функціонально залежить від вектора  $R$ :  $M_i = f(R = \{0,1\})$  у процесі аналізу матриці. Слід зауважити, що рядок матриці  $M_i$  и рядок теста  $T_i$  тут і далі є еквівалентними поняттями.

Розглядається кубітно-різницевий метод пошуку кратних дефектів на основі обчислення теоретико-множинної різниці двох векторів-рядків матриці, що відповідають об'єднанню одиничних і нульових реакцій виходів, що спостерігаються, на вхідний тест перевірки несправностей  $D_m$ :

$$D_m = \bigcup_{\forall R_i=1} M_i \setminus \bigcup_{\forall R_i=0} M_i = \bigvee_{\forall R_i=1} M_i \wedge \overline{\bigvee_{\forall R_i=0} M_i}.$$

Алфавіт опису константних несправних станів цифрової схеми [10–15] має вигляд:  $A=\{0,1,X=\{0,1\},\emptyset\}$ , де коди символів – кубіти – становлять множину  $K(A)=\{10,01,11,00\}$ . Структури даних представлені матрицею несправностей на декартовому добутку множин тестових наборів і еквіпотенційних ліній об'єкта діагностування, де кожна комірка являє собою двобітовий код-кубіт: перший ідентифікує константну несправність нуля, а другий перевіряє константну несправність. Суперпозиція несправностей (дві одиниці на одній лінії-комірці) дає можливість суттєво мінімізувати структури даних для зберігання інформації з метою подальшого пошуку дефектів під час виконання діагностичного експерименту в режимі online. Для перевірки кубітного способу пошуку дефектів далі пропонується логічна схема (рис. 3.1), яка має 6 елементів and-not, 11 ліній, 5 входів та два виходи.

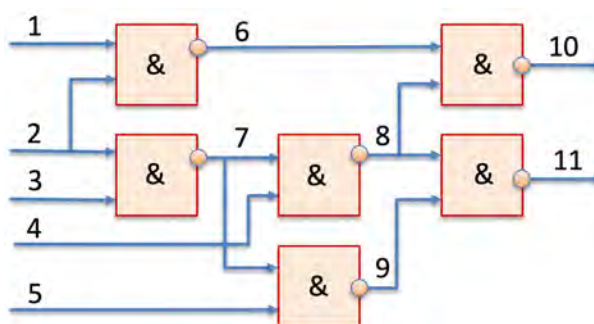


Рисунок 3.1 – Логічна схема для верифікації кубітного методу

Синтез тесту для цифрової схеми виконаний методом активізації одновимірних шляхів, кількість яких у цьому прикладі дорівнює п'яти. Отже, кількість пар тестових наборів також дорівнює п'яти:

M=<T,F>	1	2	3	4	5	6	7	8	9	10	11
T1=	0	1	1	0	0	1	0	1	1	0	0
T2=	1	1	1	0	0	0	0	1	1	1	0
T3=	0	0	1	0	1	1	1	1	0	0	1
T4=	0	1	1	0	1	1	0	1	1	0	0
T5=	1	1	0	0	1	0	1	1	0	1	1
T6=	1	1	1	0	1	0	0	1	1	1	0
T7=	1	0	1	0	1	1	1	1	0	0	1
T8=	1	0	1	1	1	1	1	0	0	1	1
T9	1	0	1	1	0	1	1	0	1	1	0
T10	1	0	1	1	1	1	0	0	1	1	1

Наступна таблиця містить повний тест для перевірки одиночних константних несправностей, які є інверсними станам справної поведінки на лініях активізації, де порожні координати відповідають порожній множині. Крім того, показаний результат виконання діагностичного експерименту для вектора експериментальної перевірки R=10100100:

M=<T,F>	1	2	3	4	5	6	7	8	9	10	11	R
T1=	1				0					1	1	1
T2=	0				1					0	1	0
T3=	1				0			1	1	0	1	
T4=	0				1			0	1	1	0	
T5=		1			0			1	0	0	0	
T6=		0			1			0	0	1	1	
T7=				1				0		1	0	0
T8=				0				1		0	0	0
T9				1				0	0	1	1	
T10				0				1	0	0	0	
$F_m^1 = \bigvee_{\forall R_i=1} T_i$	1	1	0	1	0	X		X	X	X		
$F^0 = \bigvee_{\forall R_i=0} T_i$	0	0	1	X	0	1	X	X	X	X	X	
$D_m = F^1 \wedge \overline{F^0}$	1	1	0	1	0							

Далі розглядається таблиця, одержана шляхом унітарного кодування символів несправностей [10]. Вона містить повний тест для перевірки одиночних константних несправностей, де порожні координати відповідають порожній множині, закодовані вектором 00. Показано також результат виконання діагностичного експерименту для вектора експериментальної перевірки  $R=10100100$ :

$M=\langle T,F \rangle$	1	2	3	4	5	6	7	8	9	10	11	R
T1=	01					10				01	01	1
T2=	10					01				0	01	0
T3=		01					10		01	01	10	1
T4=		10					01		10	01	01	0
T5=			01				10		01	10	10	0
T6=			10				01		10	10	01	1
T7=				01				10		01	10	0
T8=				10				01		10	10	0
T9					01				10	10	01	1
T10					0				01	10	10	0
$F_m^1 = \bigvee_{\forall R_i=1} T_i$	01	01	10		01	10	11		11	11	11	
$F^0 = \bigvee_{\forall R_i=0} T_i$	10	10	01	11	10	01	11	11	11	11	11	
$D_m = F^1 \wedge \overline{F^0}$	01	01	10		01	10						

Наступна таблиця отримана шляхом дедуктивного моделювання несправностей попереднього тесту активізації всіх логічних шляхів з його наступною мінімізацією. Таблиця несправностей ілюструє технологічні матричні структури даних, а також виконання діагностичного експерименту на основі поєднання сукупності дефектів-рядків з кубітами в комірках, які формують некоректні стани виходів на тестових наборах

{T1-R10; T5-R11; T6-(R10, R11); T8-R11}:

M=<T,F>	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	R10	R11
T1	01	10	01	00	10	00	10	00	10	10	01	1	0
T2	10	00	10	00	01	10	00	00	10	01	10	0	0
T3	00	01	01	00	00	01	10	01	01	10	10	0	0
T4	10	00	01	00	10	00	01	00	10	01	01	0	0
T5	00	10	00	01	00	01	00	10	00	10	01	0	1
T6	01	10	00	00	10	00	00	01	10	01	10	1	1
T7	01	00	00	10	00	00	01	00	10	01	01	0	0
T8	00	10	10	01	01	10	00	00	00	01	10	0	1
$F_m^1 = \bigvee_{\forall R_i=1} T_i$	01	10	11	01	11	11	10	11	10	11	11	1	1
$F^0 = \bigvee_{\forall R_i=0} T_i$	11	01	11	10	11	11	11	01	11	11	11	0	0
$D_m = F^1 \wedge \overline{F^0}$	00	10	00	01	00	00	00	10	00	00	00		
$D_m =$	.	0	.	1	.	.	.	0	.	.	.		

Тут паралельне виконання операції диз'юнкції для рядків T1, T5, T6, T8 формує вектор  $F_m^1$ , який збирає всі можливі дефекти, що перевіряються на тестових наборах. Вектор  $F^0$ , отриманий за допомогою паралельної операції диз'юнкція над рядками T2, T3, T4, T7, поєднує всі неможливі дефекти, що не перевіряються на тестових наборах. Віднімання всіх неможливих з усіх можливих дефектів дає результат у вигляді трьох константних несправностей, закодованих у таблиці як F2 = 10; F4 = 01; F8 = 10.

Таким чином, паралельне виконання двох регістрових or-операцій на основі результатів проведеного діагностичного експерименту дозволило визначити три можливі несправності, кожна з яких має місце в логічній схемі:  $D_m = \{2^0, 4^1, 8^0\}$ .

Більш жорсткою є обмежувальна умова існування в логічній схемі одиночного константного дефекту, що ймовірніше в процесі експлуатації цифрового виробу. Використання такої умови призводить до процедури пошуку дефектів на основі такого виразу:

$$D_s = \bigcap_{\forall R_i=1} M_i \setminus \bigcup_{\forall R_i=0} M_i = \bigwedge_{\forall R_i=1} M_i \wedge \overline{\bigvee_{\forall R_i=0} M_i}.$$

Застосування цієї формули суттєво уточнює результат діагностування у бік зменшення потужності можливих дефектів:  $D_s = \{2^0\}$  за рахунок отримання логічної суперечності кодів дефектів на основі використання and-операції у стовпцях 4 та 8 у таблиці, поданій нижче:

$M=\langle T,F \rangle$	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	R10	R11
T1	0	1	1	0	0	1	0	0	0	1	0	1	0
T2	1	0	0	1	0	0	0	1	0	0	1	0	0
T3	0	0	1	0	1	0	0	0	1	0	1	0	0
T4	1	0	0	0	1	0	0	0	1	0	0	1	0
T5	0	0	1	0	0	0	1	0	0	1	0	0	1
T6	0	1	1	0	0	0	0	1	0	0	1	1	1
T7	0	1	0	0	0	1	0	0	0	1	0	1	0
T8	0	0	1	0	1	0	1	0	0	0	0	1	1
$F_s^1 = \bigwedge_{\forall R_i=1} T_i$	0	0	1	0	0	0	0	0	0	0	0	1	1
$F^0 = \bigvee_{\forall R_i=0} T_i$	1	1	0	1	1	1	1	1	0	1	1	0	0
$D_s = F^1 \wedge \overline{F^0}$	0	0	1	0	0	0	0	0	0	0	0		
$D_s =$	.	0	.	.	.	.	.	.	.	.	.		

Перетин стовпців-рядків за умовами реакції схеми на тест визначає їх схожість, яка створює жорстку умову існування одиночного дефекту в об'єкті.

### 3.3 Метод еквівалентування векторів станів матриці дефектів

Розглядається метод аналізу стовпців-векторів, сутність якого полягає в еквівалентуванні множини несправних станів або синтезу класу еквівалентних дефектів за вектором експериментальних перевірок шляхом визначення метричних нормованих оцінок подібності-відмінності між парами:  $R$  і  $\exists M_j$ . Структура методу має такий вигляд (рис. 3.2).

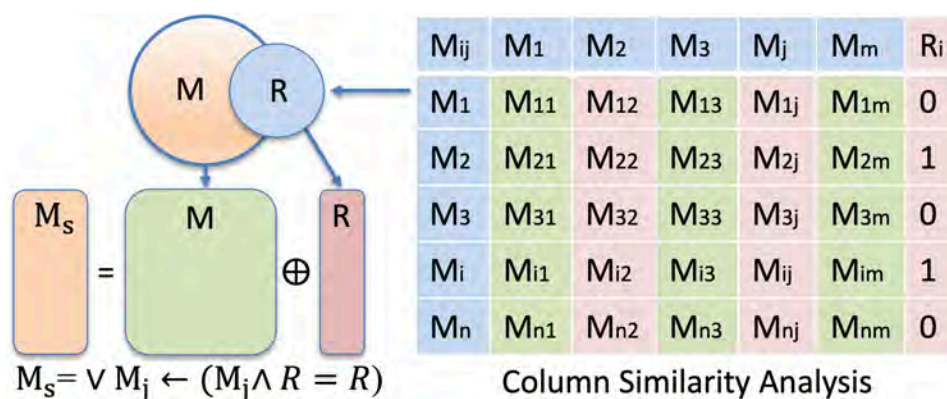


Рисунок 3.2 – Аналіз подібності по стовпцям матриці

По суті, виконується пошук множини таких стовпців-станів, які еквівалентні заданому вектору експериментальної перевірки  $R$  за критеріями максимальної (мінімальної) подібності (відмінності). Оцінки подібності-відмінності є взаємно доповнюють один одного до структурної одиниці:  $S=1-D$ ,  $D=1-S$ . Сукупність формул для обчислення структурної подібності-відмінності між вектор-стовпцями  $M_j$  двійкової матриці і  $R$ , з метою формування класу еквівалентностей ( $M_S$  or  $M_D$ ), має такий вигляд:

$$M_S = \cup M_j \leftarrow (M_j \Delta R = 0);$$

$$M_S = \cup M_j \leftarrow (M_j = R);$$

$$M_S = \cup M_j \leftarrow (M_j \cap R = R);$$

$$M_S = \cup M_j \leftarrow (M_j \oplus R = 0);$$

$$M_S = \cup M_j \leftarrow (M_j = R);$$

$$M_S = \cup M_j \leftarrow (M_j \wedge R = R);$$

$$M_D = \cup M_j \leftarrow (M_j \Delta R = 1);$$

$$M_D = \cup M_j \leftarrow (M_j \neq R);$$

$$M_D = \cup M_j \leftarrow (M_j \cap R \neq R);$$

$$M_D = \cup M_j \leftarrow (M_j \oplus R = 1);$$

$$M_D = \vee M_j \leftarrow (M_j \neq R);$$

$$M_D = \vee M_j \leftarrow (M_j \wedge R \neq R).$$

Приклад 3.1. Розглянемо використання введених формул для пошуку дефектів на основі аналізу стовпців і вектора експериментальної перевірки R. У наступній таблиці визначається схожість-відмінність тільки за істотними координатами, що створює одиничні умови порівнюваних координат по операції або:  $a \vee b = 1$  (важливо координати  $a \vee b = 0$  не вважаються суттєвими і тому не обробляються на формування подібності-відмінності). Для визначення скалярної оцінки подібності-відмінності між вектором експериментальної перевірки R і стовпцем матриці  $M_j$  використовуються простіші, похідні від наведених вище формули, що визначають метричні відносини між векторами:

$$S(R, M_j) = \frac{\sum_{i=1}^n (R_i \wedge M_{ij})}{\sum_{i=1}^n (R_i \vee M_{ij})}; \quad D(R, M_j) = \frac{\sum_{i=1}^n (R_i \oplus M_{ij})}{\sum_{i=1}^n (R_i \vee M_{ij})}.$$

Матриця несправних стовпців станів, вектора експериментальної перевірки R і скалярні нормовані оцінки подібності-відмінності представлені в таблиці:

M=<T,F>	F1	F2	F3	F4	F5	F6	F7	F8	R
T1	1		1		1				1
T2		1		1		1			0
T3	1					1			0
T4			1		1		1		1
T5	1				1				0
T6		1						1	0
T7									0
T8			1		1				1
D =	2/3	2/2	0	1/1	1/4	2/2	2/3	1	
S =	1/3	0	3/3	0	3/4	0	1/3	0	
S + D	3	2	3	1	4	2	3	1	
Faults=			1						

Підрахунок оцінок  $S(R, M_j), D(R, M_j)$  для кожного стовпця матриці дозволяє визначити подібність, представлену в рядку таблиці  $S$ , де найкращий і максимальний варіант подібності, що дорівнює трьом координатам з трьох для стовпця F3:  $S(R, M_j)=3/3$ , має нульову відмінність  $D(R, M_j)=0/3$ , рядок D.

Висновок: пристрій має несправний блок F3. Описаний підхід відрізняється від методу аналізу таблиці несправностей, запропонованого в [16], суттєвим зменшенням координат, що беруть участь у обчисленні оцінок. Стратегія пошуку дефектів при аналізі матриці несправностей використовує лише один із наступних виразів:

$$F=\cup M_j \leftarrow S(R, M_j) = \frac{\sum_{i=1}^n (R_i \wedge M_{ij})}{\sum_{i=1}^n (R_i \vee M_{ij})} = \{1 \vee \max\};$$

$$F=\cup M_j \leftarrow D(R, M_j) = \frac{\sum_{i=1}^n (R_i \oplus M_{ij})}{\sum_{i=1}^n (R_i \vee M_{ij})} = \{0 \vee \min\}.$$

При цьому слід мати на увазі, що між подібністю та відмінністю пари стовпців-векторів виконується відношення взаємного доповнення:

$$D(R, M_j) = 1 - S(R, M_j); S(R, M_j) = 1 - D(R, M_j).$$

Таким чином, сутність методу аналізу стовпців полягає в еквівалентуванні сукупності несправних станів або синтезу класу еквівалентних дефектів за вектором експериментальних перевірок шляхом визначення метричних нормованих оцінок подібності-відмінності між парами:  $R$  і  $\exists M_j$ .

### 3.4 Висновки до розділу 3

Проаналізовано методи діагностування дефектів: різницевий метод пошуку, заснований на векторному поданні множини несправностей, що перевіряються на тестових наборах, та матричний метод діагностування несправних станів цифрового виробу, що використовує як діагностичну інформацію двійкову матрицю; кубітно-різницевий метод пошуку кратних дефектів на основі обчислення теоретико-множинної різниці двох векторів-рядків матриці; метод аналізу стовпців-векторів, сутність якого полягає в еквівалентуванні сукупності несправних станів або синтезу класу еквівалентних дефектів за вектором експериментальних перевірок шляхом визначення метричних нормованих оцінок подібності-відмінності.

## 4 АПАРАТНА РЕАЛІЗАЦІЯ СЕКВЕНСОРА

Запропоновано архітектуру та виконано апаратну реалізацію подібного секвенсора для визначення включення, подібності або відмінності об'єктів. Він характеризується отриманням структурованої оцінки взаємодії двох об'єктів, що також дозволяє визначити тип трансформації одного об'єкта в інший і приймати рішення на основі суттєвості об'єктів.

### 4.1 Секвенсор для апаратного обчислення подібності, різниці та включення об'єктів

Метою є визначення якісних і кількісних метричних взаємодій між процесами та явищами за допомогою логічних паралельних регістрових операцій.

Завдання: 1) синтез логічних структур для обчислення метричних взаємодій між процесами, що подані у двійковому коді; 2) кодування структур даних і алгоритмів мовами опису апаратури; 3) тестування секвенсора, використовуючи репрезентативну вибірку тестових вхідних векторів.

Абсолютні оцінки відмінності та подібності не мають великого значення при визначенні взаємодії множин. Більш інформації надають нормовані оцінки подібності або відмінності, що наведені до суми суттєвих координат двох векторів, що виключають лише нульові за однойменними координатами значення  $X$ ,  $Y$ . Знаменник отримання нормованих оцінок виходить шляхом підсумовування всіх (поодиноких) координат після виконання диз'юнкції  $N = \sum_{i=1}^n (X_i \vee Y_i)$ . Чисельником може виступати відстань по Хеммінгу, що формує абсолютну оцінку різницю між двійковими векторами. Але далі включається модифікація, пов'язана з розподілом подібності або відмінності кількості одиничних координат, отриманих після логічного складання двійкових векторів  $X \vee Y$ . Дане доповнення робить нормовану оцінку

відмінності (подібності) значнішою за рахунок зменшення знаменника. Отже, розглядаються такі нормовані оцінки:

$$D^n = \frac{D}{N} = \frac{\sum_{i=1}^n (X_i \oplus Y_i)}{\sum_{i=1}^n (X_i \vee Y_i)}; S^n = \frac{S}{N} = \frac{\sum_{i=1}^n (X_i \wedge Y_i)}{\sum_{i=1}^n (X_i \vee Y_i)}. \quad (4.1)$$

На відміну від Хеммінга, Левенштайна та інших метрик [18-21], цей підхід полягає у видаленні суто нульових однойменних координат  $X, Y$  із структури даних з метою визначення подібності та відмінності.

Приклад 4.1. Дано два вектори (рядки), що відповідають об'єктам:

money	1	0	1	1	0	0	0	1	0	1
things	0	1	1	1	0	0	0	0	1	1

Вони повинні бути перетворені до такої мінімальної структури, де існують тільки значущі стовпці, що обов'язково міститимуть 1 за їх диз'юнкцією:

money	1	0	1	1	1	0	1
things	0	1	1	1	0	1	1

Після цього до них застосовується процедура обчислення подібності-відмінності за наведеними вище формулами (4.1):

$$D^n = \frac{4}{7} = 0,571; S^n = \frac{3}{7} = 0,429,$$

де  $D^n + S^n = 1 \rightarrow 0,571 + 0,429 = 1$ . Можна визначити похідну між нормами відмінності та подібності. При цьому шляхом віднімання чи без нього видно, яка оцінка більша – схожість чи відмінність. Тут слід обійти виникнення негативних чисел без додаткової інтерпретації. Отже, процедура обчислення оцінок подібності та відмінності приводиться до трьох векторних

паралельних операцій – сума за модулем два, диз’юнкція, кон’юнкція, після чого виконується підрахунок одиничних значень у результуючих векторах.

Якісний аналіз, який використовується для прийняття рішення про включення ключових даних до класу еквівалентності, базується на оцінці максимальної подібності або мінімальної різниці між двома наборами об’єктів. Цифрова логічна схема обчислення відмінностей та подібності показана на рис. 4.1.

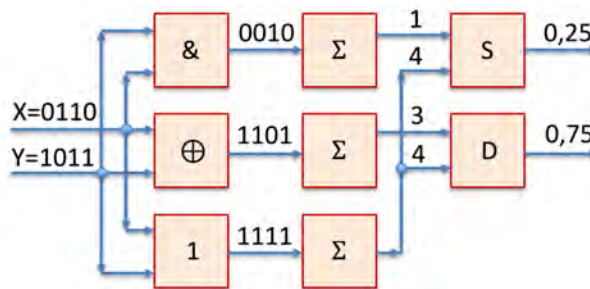


Рисунок 4.1 – Логічна схема підрахунку оцінок подібності об'єктів

Недоліком зазначеної метрики SD для визначення подібності є те, що існує нерозрізненість включення одного об’єкта. Цю недосконалість можна усунути лише за допомогою XOR вихідного вектора з результатом його перетину:

$$f(X \subseteq Y) = \frac{\sum_{i=1}^n [X_i \oplus (X_i \wedge Y_i)]}{\sum_{i=1}^n (X_i \vee Y_i)}; \quad f(Y \subseteq X) = \frac{\sum_{i=1}^n [Y_i \oplus (X_i \wedge Y_i)]}{\sum_{i=1}^n (X_i \vee Y_i)}. \quad (4.2)$$

Якщо дроб обертається на нуль, включення задовільняється:

$$X \subseteq Y \leftrightarrow \frac{\sum_{i=1}^n [X_i \oplus (X_i \wedge Y_i)]}{\sum_{i=1}^n (X_i \vee Y_i)} = 0; \quad Y \subseteq X \leftrightarrow \frac{\sum_{i=1}^n [Y_i \oplus (X_i \wedge Y_i)]}{\sum_{i=1}^n (X_i \vee Y_i)} = 0. \quad (4.3)$$

Одночасна наявність нуля в обох виходах свідчить про рівність векторів  $X=Y$ . На рис. 4.2 показана модифікована логічна схема обчислення значень подібності об'єктів, яка дозволяє визначити зв'язки включення.

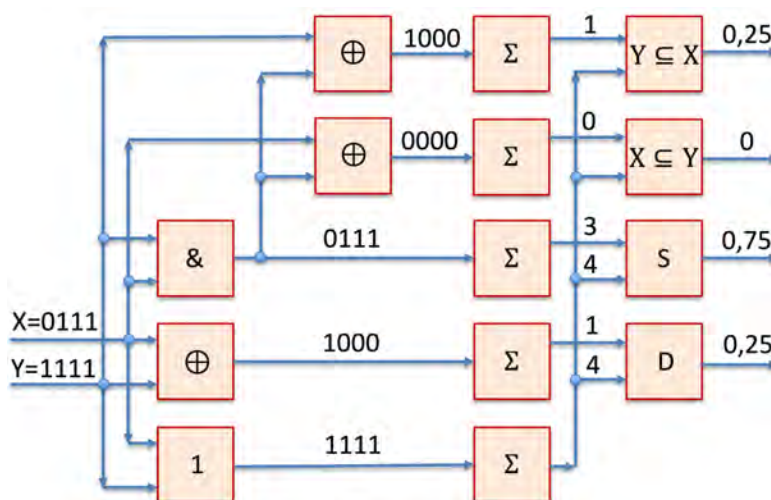


Рисунок 4.2 – Модифікована логічна схема обчислення значень подібності об'єктів для визначення включення

Важливим є не лише загальна симетрична оцінка різниці та подібності в метриці векторів, вирівняних за довжиною, але й те, який відсоток кожного ключового вектора враховується оцінкою, що формує асиметрію зв'язку. Асиметричний зв'язок як відношення дає змогу визначити пріоритетність двох векторів і вибрати, який із них є домінуючим.

Приклад 4.2. Дано два вектори  $X=1011111$ ,  $Y=0111110$ . Нехай ідентифікатором їхньої спільності виступає вектор  $X$ , оскільки вага його суттєвих одиничних координат відмінності більша, ніж у векторі  $Y$ :  $D1(X) = 2 > D1(Y) = 1$  за наявності домінуючої ненульової подібності (непустого перетину) над інтегральною відмінністю двох векторів  $S(X) = 4 > D(Y) = 3$ . Як результат записуються два об'єднані дані як конкатенація та вектори об'єднуються. Перевага такого модифікованого методу вимірювання різниці та подібності полягає в тому, що ключові вектори мають однакову довжину.

Це пов'язано з метрикою Левенштейна, яка стверджує, що нульових координат (векторів-стовпців) з однаковою назвою не існує і не може існувати. Внаслідок цього оцінки абсолютної та відносної подібності-відмінності слід вважати абсолютно симетричними, але не асиметричними в загальному випадку кількості одиниць у векторі відносно загальної довжини векторів:

$$X = \text{Image} = 0111111;$$

$$Y = \text{Prototype} = 1011110.$$

4.2. Синтез SD-автомата для виявлення подібності або відмінності об'єктів

Структура секвенсора для визначення оцінки подібності, різниці чи включення показана на рис. 4.3, де існує чотири рівні перетворення даних

- 1) формування вхідних даних по об'єктах;
- 2) синтез загальних метрик для об'єктів;
- 3) унитарне кодування об'єктів у метриці;
- 4) подання відповідного об'єкту вектору на вхід автомата SD;
- 5) формування чотирьох вихідних значень, що визначають зв'язки між об'єктами як відношення.

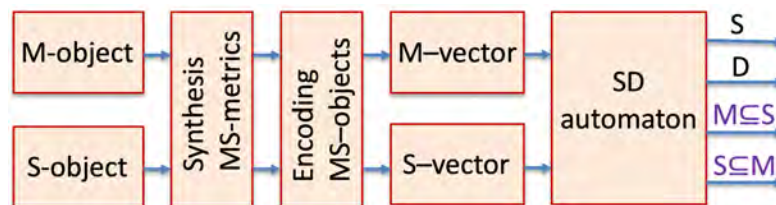


Рисунок 4.3 – Схема секвенсора для визначення відношень між об'єктами

На схемі (рис. 4.4) показано структуру пристрою для визначення різниці схожості між двома об'єктами, реалізованого у вигляді автомата SD (Similarity-Difference) на мові опису апаратури Verilog.

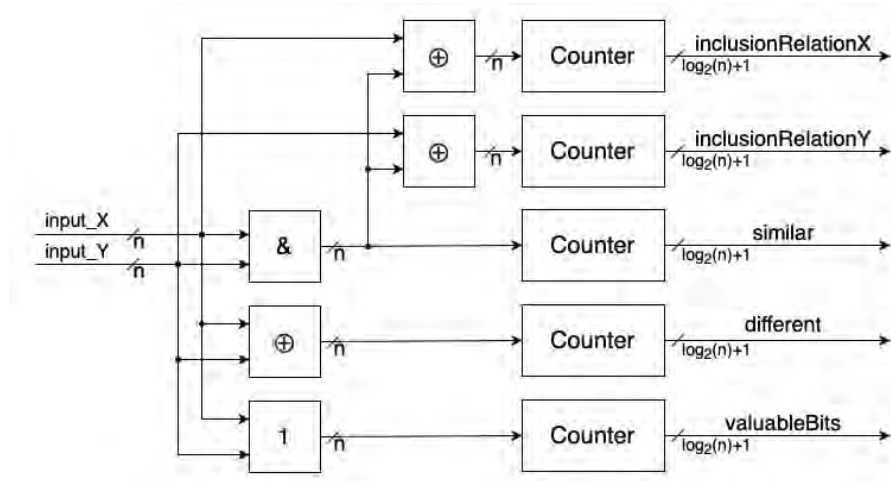


Рисунок 4.4 – Структурна схема SD-автомата

Пристрій виконує підрахунок кількості збіжних ненульових бітів (similar), бітів (different), суттєвих координат (valuableBits), а також ступеня входження  $X$  і  $Y$  (inclusionRelationX і inclusionRelationY відповідно). Координата вважається суттєвою, якщо одиниця з'являється принаймні в одному з вхідних векторів у бітах після цієї координати. Для точних обчислень подібності/відмінності нульові координати  $X$  і  $Y$  з однаковими назвами видаляються, і враховуються лише ненульові координати. Підрахунок кількості відповідних ненульових бітів виконується у два етапи. Перший крок виконує операцію "&". Потім отриманий вектор відповідних ненульових бітів подається в лічильник і підраховується кількість одиниць у цьому векторі. Лічильники можуть бути реалізовані різними способами.

Перший – за допомогою комбінаційної схеми. У цьому випадку одиниці підраховуються на логічній схемі, а затримка між впливом вхідного і вихідного результату формується найдовшим ланцюгом логічних елементів

комбінаційної схеми. Такий варіант дозволяє максимально швидко обчислити результати, але він дорогий з точки зору апаратури.

Другий спосіб полягає у використанні лічильника, який тактується зовнішнім пристроєм і використовує тригер. Цей варіант потребує мінімум обладнання, але й більше часу для досягнення результату.

Третій спосіб заснований на поєднанні двох вказаних вище варіантів. Можна порахувати одиниці у векторі за допомогою лічильника в тригері, який містить комбінаційну схему. Це дозволяє перевіряти кілька координат за автоматний такт, скорочуючи час для отримання результату.

Варіанти можна використовувати в поєднанні для досягнення еквівалентності між допустимими розмірами пристрою та часом подібних обчислень різниці або подібності. Насправді всі варіанти життєздатні. Це залежить від обмежень і потреб ринку чи споживача, для якого цей модуль розроблений.

Щоб обчислити показники подібності та несхожості в діапазоні від 0 до 1, кількість схожих і різних чисел потрібно розділити на кількість відповідностей. Залежно від специфікації проекту цю операцію можна виконувати як у цьому модулі, так і в блоці обробки (процесорі). Розроблений модуль є параметризованим і може використовуватися для вхідних векторів довільного розміру.

У таблиці 4.1 надано лістинг верифікованого програмного модуля для обчислення різниці чи подібності між двома метрично параметризованими об'єктами.

Таблиця 4.1 – Лістинг програмного модуля

```

module tb();

localparam INPUT_WIDTH = 16;

localparam OUTPUT_WIDTH = $clog2(INPUT_WIDTH)+1;

logic [INPUT_WIDTH-1 : 0] input_X;
logic [INPUT_WIDTH-1 : 0] input_Y;
logic [OUTPUT_WIDTH-1 : 0] similarity;
logic [OUTPUT_WIDTH-1 : 0] difference;
logic [OUTPUT_WIDTH-1 : 0] valuebleBits;
logic [OUTPUT_WIDTH-1 : 0] inclusionRelationX;
logic [OUTPUT_WIDTH-1 : 0] inclusionRelationY;
combinationalCircuit #(

.INPUT_WIDTH(INPUT_WIDTH)

) i_combinationalCircuit (
.input_X(input_X),
.input_Y(input_Y),
.similarity(similarity),
.difference(difference),
.valuebleBits(valuebleBits),
.inclusionRelationX(inclusionRelationX),
.inclusionRelationY(inclusionRelationY)
);

initial
begin

    input_X <= 'b0000000000000000;
input_Y <= 'b0000000000000000;
    #1;
input_X <= 'b1111111111111111;
input_Y <= 'b0000000000000000;
    #20;
input_X <= 'b1111001111111111;
input_Y <= 'b0000001100000000;
    #20;
input_X <= 'b1000011111111111;
input_Y <= 'b00011111111111000;
    #20;
input_X <= 'b1011001111111111;
input_Y <= 'b000111100011000;
    #20;
input_X <= 'b1011001111000011;
input_Y <= 'b000111101011000;
    #20;
input_X <= 'b1111111111000011;
input_Y <= 'b000111100000000;
    #20;
input_X <= 'b1100011111000011;
input_Y <= 'b0110111100111111;
end

    initial
begin
# 250;
$finish;
End

initial
begin
$dumpfile("dump.vcd");
$dumpvars;

End

endmodule

```

```

module countOnes
#(
parameter INPUT_WIDTH = 16,
localparam OUTPUT_WIDTH = $clog2(INPUT_WIDTH)+1
)
input [INPUT_WIDTH-1 : 0] input_data,
output reg [OUTPUT_WIDTH-1 : 0] numberOfOnes
);
// Combinational calculation of number of ones
always @(input_data) begin
numberOfOnes = 0;
foreach(input_data[i])
if (input_data[i] == 1'b1)
numberOfOnes = numberOfOnes + 1;
end
endmodule

module combinationalCircuit #(
parameter INPUT_WIDTH = 16,
localparam OUTPUT_WIDTH = $clog2(INPUT_WIDTH)+1
)
input [INPUT_WIDTH-1 : 0] input_X,
input [INPUT_WIDTH-1 : 0] input_Y,
output [OUTPUT_WIDTH-1 : 0] similarity,
output [OUTPUT_WIDTH-1 : 0] difference,
output [OUTPUT_WIDTH-1 : 0] valuebleBits,
output [OUTPUT_WIDTH-1 : 0] inclusionRelationX,
output [OUTPUT_WIDTH-1 : 0] inclusionRelationY
);
reg [INPUT_WIDTH-1 : 0] xorXY;
reg [INPUT_WIDTH-1 : 0] andXY;
reg [INPUT_WIDTH-1 : 0] orXY;
reg [INPUT_WIDTH-1 : 0] inclusionX;
reg [INPUT_WIDTH-1 : 0] inclusionY;
assign xorXY = input_X ^ input_Y;
assign andXY = input_X & input_Y;
assign orXY = input_X | input_Y;
assign inclusionX = input_X^andXY;
assign inclusionY = input_Y^andXY;
countOnes #(
.INPUT_WIDTH(INPUT_WIDTH)
) i_countOnes_xor (
.input_data(xorXY),
.numberOfOnes(difference)
);
countOnes #(
.INPUT_WIDTH(INPUT_WIDTH)
) i_countOnes_and (
.input_data(andXY),
.numberOfOnes(similarity)
);
countOnes #(
.INPUT_WIDTH(INPUT_WIDTH)
) i_countOnes_or (
.input_data(orXY),
.numberOfOnes(valuebleBits)
);
countOnes #(
.INPUT_WIDTH(INPUT_WIDTH)
) i_countOnes_inclusionX (
.input_data(inclusionX),
.numberOfOnes(inclusionRelationX)
);
countOnes #(
.INPUT_WIDTH(INPUT_WIDTH)
) i_countOnes_inclusionY (
.input_data(inclusionY),
.numberOfOnes(inclusionRelationY)
);
Endmodule

```

## 4.2 Валідація апаратної реалізації

На часових діаграмах (рис. 4.5) подано фрагменти моделювання у Verilog коді для валідації всіх суттєвих функцій та станів SD-автомата.

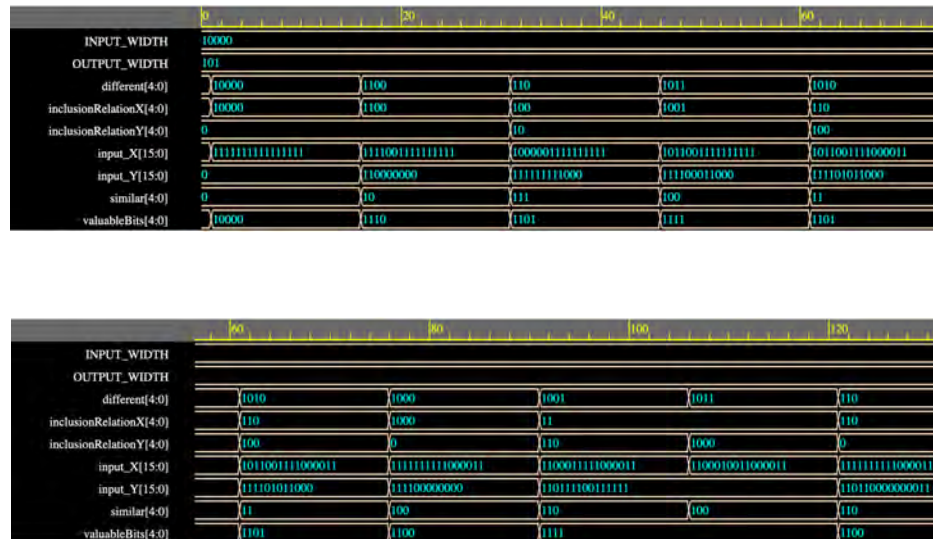


Рисунок 4.6 – Часові діаграми для валідації апаратної реалізації SD-автомата

## 4.3 Верифікація

Програмний SD-модуль верифіковано на кількох десятках тестових об'єктів, які були трансформовані до двійкових векторів з метою подальшої паралельної обробки на синтезованому цифровому автоматі.

У таблиці 4.2 наведено дані та на рис. 4.6 – відповідні графіки подібності, відмінності та включення для сукупності з десяти пар об'єктів.

Таблица 4.2 – Вихідні дані

Master, bit	45	89	36	56	77	21	22	18	12	77
Slave, bit	47	72	31	65	78	37	45	11	21	67
Similarity	0,9	0,8	0,6	0,7	1,0	0,5	0,6	0,0	0,8	0,3
Difference	0,1	0,2	0,4	0,3	0,0	0,5	0,4	1,0	0,2	0,7
MSC	1	0	0	0	1	0	1	0	0	0
SMC	0	0	1	0	1	0	0	0	1	0

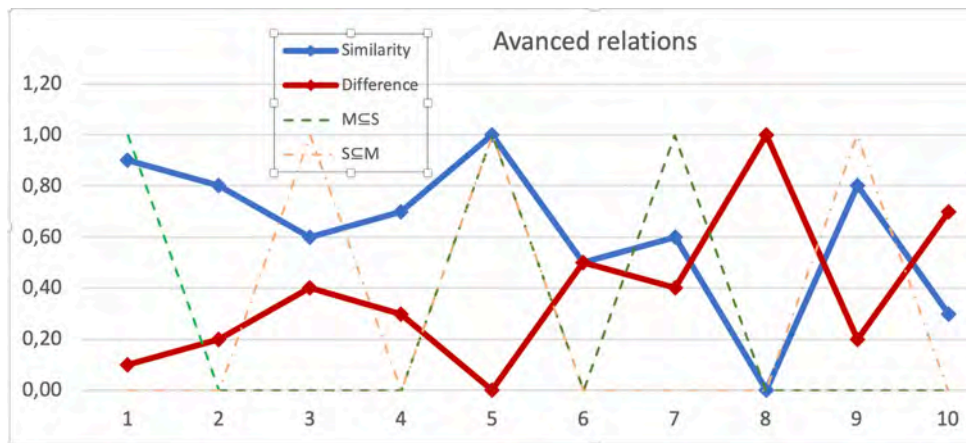


Рисунок 4.6 – Графіки щодо верифікації секвенсора

З графіків (див. рис. 4.6) слідує, що симетрія відношення подібності-відмінності завжди підтримується (підтверджується) графіком відношення включення, і насправді зв'язки між об'єктами майже завжди асиметричні. Два верхніх рядки таблиці вказують ряд параметрів, важливих для ідентифікації об'єкта. У той же час ці різні числа приводяться до загальної метрики шляхом об'єднання окремих метрик, і обчислюються чотири вихідні значення автомата для визначення подібності та відмінності. Останні чотири рядки формують структуру взаємодії двох об'єктних векторів та їх оцінок. Оцінка враховує не лише подібність і несхожість, але й показники включення, які неочевидні зі значень подібності об'єктів. Апаратні секвенсори пошуку подібних відмінностей можна використовувати для визначення структури зв'язків між об'єктами, представленими чотирма типами взаємодії, показаними на рис. 4.7.

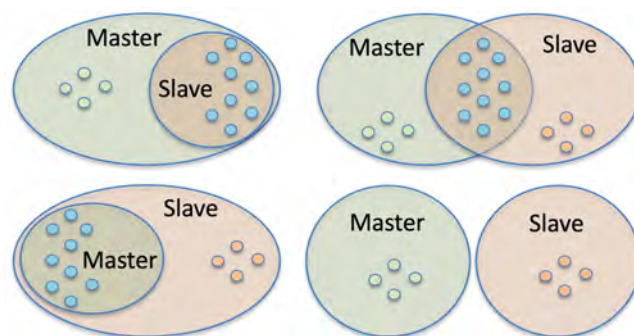


Рисунок 4.7 – Діаграми відношень між об'єктами для SD-автомата

#### 4.4 Висновки до розділу 4

Адаптована логічна метрика якості розпізнавання патернів, дефектів та колізій, яка разом із рівнянням комп'ютингу дає можливість формувати всі структурні та нормовані оцінки в процесі навчання.

Запропонована система розпізнавання патернів та прийняття рішення верифікована на представницькій вибірці тестових впливів та може бути використана на ринку цифрових технологій.

Запропоновано апаратну реалізацію пошукового секвенсора для виявлення схожості, різниці та включення. Він характеризується отриманням структурованої оцінки взаємодії двох об'єктів, що дозволяє більш точно визначати та вибирати суттєві об'єкти шляхом перетворення одного об'єкта в інший.

## ВИСНОВКИ

У магістерській роботі було розглянуто питання, пов'язані зі створенням моделей, методів, архітектур на основі матричних структур даних для пошуку дефектів у цифрових системах.

Для досягнення мети дослідження – суттєвого зменшення обчислювальної складності алгоритмів пошуку даних та діагностування несправностей шляхом розробки ефективної інфраструктури матричних даних для технологічного апаратно-орієнтованого паралельного аналізу дефектів з використанням метрики подібності-відмінності – в роботі вирішено наступні задачі:

- 1) проаналізовано сучасні технологічні тенденції;
- 2) виконано аналітичний огляд математичного апарату – моделей та методів пошуку та діагностування дефектів;
- 3) розроблено архітектуру секвенсору для пошуку подібності-відмінності-включення на основі матричних структур даних.

Адаптована логічна метрика якості розпізнавання патернів, дефектів та колізій, яка разом із рівнянням комп'ютингу дає можливість формувати всі структурні та нормовані оцінки в процесі навчання.

Система розпізнавання патернів та прийняття рішення верифікована на представницькій вибірці тестових впливів та може бути використана на ринку цифрових технологій.

Запропоновано апаратну реалізацію пошукового секвенсора для виявлення схожості, різниці та включення. Він характеризується отриманням структурованої оцінки взаємодії двох об'єктів, що дозволяє більш точно визначати та вибирати суттєві об'єкти шляхом перетворення одного об'єкта в інший.

Таким чином, новизна архітектури секвенсору пошуку подібності-відмінності-включення полягає в отриманні структурованої оцінки взаємодії

двох об'єктів, яка дає можливість точніше визначати шляхи перетворення одного об'єкта до іншого, а також вибирати суттєвіші об'єкти з пари при прийнятті рішення.

Обмеження методів пошуку даних пов'язані з необхідністю виконання процедури синтезу матричних структур даних, що має квадратичну обчислювальну складність. Подальші дослідження можуть бути спрямовані на створення програмно-апаратного продукту, що реалізуватиме паралельні процедури пошуку даних у метричному матричному просторі на основі заданого вектора вхідних умов.

Результати відображено у тезах доповіді на науковій конференції [22] (додаток Б).

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Samuel K. Moore. A Better Way To Measure Progress In Semiconductors: It's time to throw out the old Moore's Law metric. 21 JUL 2020. Режим доступу: <https://spectrum.ieee.org/a-better-way-to-measure-progress-in-semiconductors>

2. Dexter Johnson. IBM's \$3-Billion Research Project Has Kept Computing Moving Forward. With IBM's "7-nm and Beyond" research initiative completed, it's time to assess how its innovations have helped chipmakers keep up with the demands of Moore's Law. 14 JUL 2020. Режим доступу: <https://spectrum.ieee.org/ibms-3billion-research-project-has-kept-computing-moving-forward>

3. Samuel K. Moore. 96-Core Processor Made of Chiplets: CEA-Leti bound dozens of processor cores together by breaking them up and stitching them back together on a separate piece of silicon. 18 FEB 2020. Режим доступу: <https://spectrum.ieee.org/tech-talk/semiconductors/processors/core-processor-chiplets-isscc-news>

4. Samuel K. Moore. Edge-AI Startup Gets \$60-million, Preps for Mass Production: Hailo sped from an idea to a 26-trillion-operations-per-second chip in three years. 12 MAR 2020. Режим доступу: <https://spectrum.ieee.org/tech-talk/semiconductors/processors/israeli-edgeai-startup-gets-60million-preps-for-mass-production>

5. Samuel K. Moore. Cerebras's Giant Chip Will Smash Deep Learning's Speed Barrier: Computers using Cerebras's chip will train these AI systems in hours instead of weeks. 01 JAN 2020. Режим доступу: <https://spectrum.ieee.org/semiconductors/processors/cerebrass-giant-chip-will-smash-deep-learning-speed-barrier>

6. Tekla S. Perry. How the Father of Finfets Helped Save Moore's Law: Chenming Hu, the 2020 IEEE Medal of Honor recipient, took transistors into the third dimension. 21 APR 2020. Режим доступу:

<https://spectrum.ieee.org/semiconductors/devices/how-the-father-of-finfets-helped-save-moores-law>

7. Charles Q. Choi. Image Sensor Doubles as a Neural Net: The device analyzes images thousands of times faster than conventional machine-vision strategies. 04 MAR 2020. Режим доступу: <https://spectrum.ieee.org/tech-talk/computing/hardware/image-neural>

8. Abramovici M. Digital System Testing and Testable Design / M. Abramovici, M.A. Breuer and A.D. Friedman. Comp. Sc. Press. 1998. 652 p.

9. M. Karavay, V. Hahanov, E. Litvinova, H. Khakhanova and I. Hahanova, "Qubit Fault Detection in SoC Logic," 2019 IEEE East-West Design & Test Symposium (EWDTS), Batumi, Georgia, 2019, pp. 1-7.

10. Vladimir Hahanov. Cyber Physical Computing for IoT-driven Services New York: Springer 2018. 279 p.

11. Hahanov V.I., Tamer Bani Amer, Chumachenko S.V., Litvinova E.I. Qubit technology analysis and diagnosis of digital devices // Electronic modeling. 2015. Vol. 37, no 3. P. 17-40.

12. Hahanov V., Litvinova E., Gharibi W., Chumachenko S. Big Data Driven Cyber Analytic System // IEEE International Congress on Big Data, New York City. 2015. P. 615-622.

13. Hahanov I., Gharibi W., Iemelianov I., Bani Amer T. QuaSim – Cloud Service for Digital Circuits Simulation // Proceedings of IEEE East-West Design & Test Symposium. – 2016. – Yerevan, Armenia. – P. 363-370.

14. Хаханов В.І., Бондаренко М.Ф., Литвинова Є.І. Структура логічного асоціативного мультипроцесора // Автоматика і телемеханіка. 2012. № 10. С. 73-94.

15. Vladimir Hahanov; Svetlana Chumachenko; Eugenia Litvinova; Abdullayev Vugar Nacimahmud; Anastasia Hahanova; Tetiana Soklakova. Cyber Social Computing. 2018 IEEE East-West Design & Test Symposium (EWDTS). 2018.

16. Malyshenko Yu.V. Technical diagnostics. VGUES Publishing House, 2010. 300 p.

17. Fernandes de Mello Rodrigo, Moacir Antonelli Ponti. Machine Learning. Practical Approach on the Statistical Learning Theory. Springer. 2018. 362 p.

18. Tanimoto T.T., An elementary mathematical theory of classification and prediction, IBM Report. (November 1958), cited in: G. Salton, Automatic Information Organization and Retrieval (McGraw-Hill, 1968) p. 238.

19. Temma S., Sugii M. and Matsuno H., "The Document Similarity Index based on the Jaccard Distance for Mail Filtering," 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), JeJu, Korea (South), 2019, pp. 1-4.

20. Verma N. K., Dutta E. and Yan Cui, "Hausdorff distance and global silhouette index as novel measures for estimating quality of biclusters," 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Washington, DC, 2015, pp. 267-272.

21. Jaccard P. Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines // Bull. Soc. Vaudoise sci. Natur. – 1901. – V. 37, Bd. 140. – S. 241-272.

22. Каряка Е.В. Діагностування дефектів у кіберфізичних системах / 27-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 5. – Харків: ХНУРЕ. 2023. – С. 103-104.