

ВЕКТОРНІ МОДЕЛІ ЛОГІКИ І СТРУКТУРИ ДЛЯ ТЕСТУВАННЯ ТА МОДЕЛЮВАННЯ ЦИФРОВИХ СХЕМ

Хаханова А. В. – канд. техн. наук, доцент каф. Автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Хаханов В. І. – д-р техн. наук, професор каф. Автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Чумаченко С. В. – д-р техн. наук, професор, зав. каф. Автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Литвинова Є. І. – д-р техн. наук, професор каф. Автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Рахліс Д. Ю. – канд. техн. наук, доцент каф. Автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

АНОТАЦІЯ

Актуальність. Відомо, що структури даних є визначальними для створення ефективних паралельних алгоритмів і високопродуктивних обчислювальних пристроїв. Тому розробка математично досконалих і технологічно простих структур даних займає близько 80 відсотків часу проектування, коли на алгоритми і їх hardware-software кодування витрачається близько 20 відсотків часових і матеріальних ресурсів. Це обумовлює пошук таких примітивів структур даних, які суттєво спростять паралельні високопродуктивні алгоритми, що працюють на них. Пропонуються моделі і методи для тестування та моделювання цифрових систем, що містять окремі переваги квантового комп'ютингу в частині імплементації векторних кубітних структур даних в технології класичних обчислювальних процесів.

Мета. Розробка інноваційної технології кубітно-векторного синтезу і дедуктивного аналізу тестів для їх верифікації на основі векторних структур даних, що істотно спрощують алгоритми, які можуть бути вбудовані як компоненти BIST в цифрові системи на кристалах.

Метод. Використовується дедуктивне моделювання несправностей для отримання аналітичних виразів, орієнтованих на транспортування списків несправностей через функціональний або логічний елемент на основі хог-операції, яка виконує роль вимірника подібності-відмінності між тестом, функцією і несправностями, заданими однаково в одному з форматів – таблицю, графом, рівнянням. Пропонується двійковий вектор як самий технологічний примітив структур даних для завдання логічної функціональності з метою паралельного синтезу та аналізу цифрових систем. Паралелізм рішення комбінаторних задач є фізична властивість квантового комп'ютингу, що в класичному комп'ютингу, для паралельного моделювання та діагностування несправностей, забезпечується унітарно-кодованими структурами даних, завдяки надлишкової пам'яті.

Результати. 1) Розроблено метод аналітичного синтезу дедуктивної логіки для функціональних елементів вентильного рівня і рівня регістрових передач. 2) Запропоновано дедуктивний процесор для моделювання несправностей на основі транспортування вхідних списків або векторів несправностей на зовнішні виходи цифрових схем. 3) Описано кубітно-векторну форму завдання логіки та методи кубітного синтезу дедуктивних рівнянь для моделювання несправностей. 4) Розроблено кубітно-векторний метод синтезу тестів, що використовує похідні, які обчислюються за векторним покриттям логіки. 5) Виконано верифікацію моделей і методів на тестових прикладах в програмній реалізації структур і алгоритмів.

Висновки. Наукова новизна полягає в новій парадигмі технології синтезу дедуктивної RTL-логіки на основі метричного рівняння тестування, що формує хог-відносини між тестом, функцією і несправностями. Вводиться векторна форма опису структур, яка дає можливість застосувати відомі технології синтезу та аналізу тестів логічних схем для ефективного вирішення завдань тестування графових структур і автоматичних моделей цифрових пристроїв. Практична значимість відбивається в прикладах аналітичного синтезу дедуктивної логіки для функціональних елементів векторного рівня і рівня регістрових передач. Пропонується дедуктивний процесор для моделювання несправностей, який орієнтований на імплементацію як засобу BIST, що використовується в режимі онлайн тестування, моделювання і діагностування несправностей для цифрових систем на кристалах. Пропонується кубітно-векторна форма опису цифрових систем, яка перевершує існуючі способи завдання обчислювальних пристроїв за метрикою: технологічність, компактність, швидкодія і якість. Розроблено програмний застосунок, який реалізує основні сервіси тестування, моделювання та діагностування, що використовуються в навчальному процесі для вивчення переваг кубітно-векторних структур даних і алгоритмів. Наводяться оцінки обчислювальної складності процесів синтезу тестів і дедуктивних формул для логіки і їх використання при моделюванні несправностей.

КЛЮЧОВІ СЛОВА: RTL логіка, синтез, технічна діагностика, моделювання цифрових схем, X-функція, метрика подібності-відмінності, дедуктивна функція, метрика рівняння тестування.

АБРЕВІАТУРИ

AI – штучний інтелект;

BIST – вбудоване самотестування;

Design – проектування;

FC – fault coverage (покриття несправностей);

FD – fault detected (виявлені несправності);

ML – машинне навчання;

RTL – register transfer level (рівень регістрових передач);

SoC – система на кристалі;

Testing – тестування;

АЛП – арифметико-логічний пристрій;

ДНФ – діз'юнктивна нормальна форма.

НОМЕНКЛАТУРА

$C = (C_1, C_2, \dots, C_i, \dots, C_n)$ – таблиця істинності;

$C_i = \{X_i Y_i\}$ – сукупністю вхідних двійкових векторів X_i , кожному з них ставиться у відповідність стан виходу Y_i ;

$D(a, b)$ – метрика відмінності двох об'єктів a і b ;

d_i – відстань Хеммінга між двома моделями об'єктів;

F – функція;

G – орієнтований граф;

H – матриця перестановки;

k – число двійкових змінних;

L – несправність;

L_{ij}^* – дедуктивна матриця для паралельного аналізу

одиначних константних несправностей;

n – розмірність таблиці істинності;

N – кількість вершин графа;

$N_1 (N_0)$ – число вершин в області відправлення (прибуття) розглянутої відповідності;

Q -логіка – логіка, що оперує кубітами (векторними формами унітарного кодування універсуму з n примітивів);

Q -покриття – вектор станів вихідної змінної таблиці істинності;

$\{Q, E, H, J\}$ – символи універсума примітивів, що генерують булеан;

$S(a, b)$ – метрика подібності двох об'єктів a і b ;

T – тест;

T_{i1}, T_{i2} – значення відповідних вхідних змінних на тестових наборах;

U – число змінних управління;

V_i – вершина графа;

V_g – опису графа;

V_t – завдання тесту;

V_f – вектор несправностей;

X_i – вхідний двійковий вектор;

X_1, X_2 – вхідні списки або вектори несправностей, які транспортуються на вихід елемента;

$Y = (Y_1, Y_2, \dots, Y_i, \dots, Y_n)$ – вектор станів виходу логічного елемента;

Y_i – стан вихідної змінної на i -наборі тесту.

ВСТУП

Парадигма створення квантового ринкового комп'ютера, що використовує квантову логіку на основі кубітних структур даних, є сумнівною. Метричні підстави: 1) Q -логіка, як і звичайна, створює до 90 відсотків всіх проблем, пов'язаних з технологією її апарат-

ної реалізації, продуктивністю, якістю, надійністю, тестуванням, діагностуванням та ремонтом. 2) Включення класичної та квантової логіки з комп'ютера дає можливість перейти на більш високий рівень архітектури комп'ютерингу, яка використовує тільки регулярну пам'ять для зберігання і логічного перетворення даних. 3) При цьому усуваються недоліки, пов'язані з використанням обов'язкових сьогодні блоків АЛП та шин даних, які в десятки разів сповільнюють обчислювальні процеси. 4) Перспективною стає архітектура Q -комп'ютера, що використовує тільки структуру електронів як пам'ять і транзакції між ними на основі керованих фотонів. 5) Швидкодія звернення до електрон-пам'яті і виконання транзакцій на ній визначається швидкістю світла, що робить такий Q -комп'ютер ринково привабливим і унікальним для вирішення комбінаторних завдань. 6) Технічні і температурні умови для реалізації квантового обчислювача є поки непереборним обмежувачем для його масової імплементації в існуючий кіберпростір.

Проте, далі пропонуються моделі і методи для тестування і моделювання цифрових систем, що містять окремі переваги квантового комп'ютерингу в частині імплементації векторних кубітних структур даних в технології класичних обчислювальних процесів. Відомо, що структури даних є визначальними для створення ефективних паралельних алгоритмів і високопродуктивних обчислювальних пристроїв. Тому розробка математично досконалих і технологічно простих структур даних в «розумних» компаніях займає близько 80 відсотків часу проектування, коли на алгоритми та їх hardware-software кодування витрачається близько 20 відсотків часових і матеріальних ресурсів.

Звідси випливає простий висновок – необхідно знайти такі примітиви структур даних, які суттєво спростять паралельні високопродуктивні алгоритми, що працюють на них. Виходячи зі сказаного, далі пропонується двійковий вектор, як самий технологічний примітив структур даних для завдання логічної функціональності з метою паралельного синтезу та аналізу цифрових систем. Паралелізм рішення комбінаторних задач є фізична властивість квантового комп'ютерингу, яке в класичному комп'ютерингу, для паралельного моделювання несправностей (parallel fault simulation) і діагностики несправностей (fault diagnosis), забезпечується унітарно кодованими структурами даних, завдяки надлишковій пам'яті [1].

Об'єкт дослідження – процес паралельного синтезу та аналізу цифрових систем.

Предмет дослідження – методи кубітно-векторного синтезу і дедуктивного аналізу тестів та їх верифікація на основі векторних структур даних.

Мета дослідження – запропонувати інноваційну технологію кубітно-векторного синтезу і дедуктивного аналізу тестів для їх верифікації на основі векторних структур даних, що істотно спрощують алгоритми, які можуть бути вбудовані як компоненти BIST в цифрові системи на кристалах.

1 ПОСТАНОВКА ЗАДАЧІ

Інноваційна ідея для вирішення проблем тестування. Ринкова метрика квантового комп'ютингу – паралелізм розв'язання комбінаторних задач, завдяки структурам даних, де примітивом є кубіт, що одночасно має два стани $X = \{0,1\}$. Ці стани можна унітарно закодувати для класичного комп'ютера як суперпозицію – об'єднання двох символів $11 = \{10, 01\}$ відповідно. Природно, можна придумати будь-яке кінцеве число символів, що одночасно створюють один, більш складний чотирьохкомпонентний кубіт-стан, наприклад, $Y = \{Q, E, H, J\} \rightarrow 1111 = \{1000, 0100, 0010, 0001\}$ [18, 23, 24, 25, 27, 29, 30]. Застосування кубітності (векторності) до логіки також має сенс паралелізму. Таблицю істинності хог-елемента можна зобразити, наприклад, у вигляді унітарних кодів вхідних наборів: $00\ 0\ 01\ 1, 10\ 1, 11\ 0 \rightarrow 1000\ 0, 0100\ 1, 0010\ 1, 0001\ 0$. Дані коди можна попарно (1,4 і 2,3) суперпозиціонувати або мінімізувати до наступного вигляду: $1001, 0, 0110\ 1$. Природно, досить мати тільки один з них для ідентифікації функції (структури), зважаючи на їх додатковості, наприклад, $0110\ 1$. Це є не тільки результат мінімізації, але ще й короткий запис ДНФ. Відкинувши останню одиницю, дедуктивно виходить вектор вихідних станів 0110 , який тут і далі ідентифікується або визначається кубітним.

Для досягнення поставленої мети треба вирішити наступні задачі:

1) розробити метод аналітичного синтезу дедуктивної логіки для функціональних елементів вентилярного рівня (gate level) і рівня регістрових передач (register transfer level) [25];

2) запропонувати дедуктивний процесор для моделювання несправностей на основі транспортування вхідних списків або векторів несправностей на зовнішні виходи цифрових схем [27];

3) описати кубітно-векторну форму завдання логіки та методи кубітного синтезу дедуктивних рівнянь для моделювання несправностей [18];

4) розробити кубітно-векторний метод синтезу тестів, що використовує похідні, які обчислюються за векторним покриттям логіки [29];

5) виконати верифікацію моделей і методів на тестових прикладах в програмній реалізації структур і алгоритмів [24].

2 ОГЛЯД ЛІТЕРАТУРИ

Тематика статті викликає певний інтерес з боку дослідників, що займаються тестуванням проектів [1] і верифікацією цифрових систем на кристалі з точки зору технологій наноелектроніки, що з'являються. Субатомний рівень деталізації мікросхем вимагає розробки нових підходів в області синтезу тестів, моделювання несправностей і діагностики з метою створення тестопригодних і відмовостійких обчислювальних систем [2–17]. Це підтверджується численними публікаціями в області технологій проектування і тестування. а саме: в бібліотеці IEEE Xplore по даній темі

знайдено 175920 публікацій, серед яких напрямок, пов'язаний з моделюванням несправностей, становить близько 25% (43028) опублікованих робіт. Крім того, відповідна оцінка двох згаданих областей досліджень була зроблена видавництвом Springer, де опубліковано відповідно 182717 і 15603 монографій. Класифікація технологій моделювання (Fault Simulation) несправностей може бути представлена відношенням структур даних, моделей (Models), методів (Methods) і типів несправностей (Fault models) (рис. 1) [18].

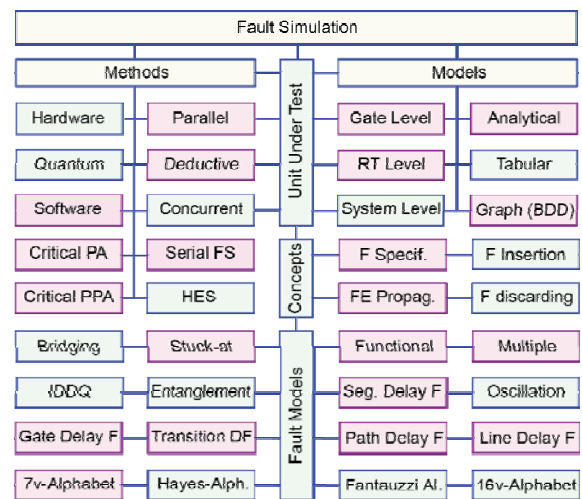


Рисунок 1 – Класифікація технологій моделювання несправностей

Згідно з рисунком 1, блок, що тестується (Unit Under Test), може бути представлений моделями вентилярного рівня (Gate Level) чи рівня регістрових передач (RT Level), аналітичними (Analytical), табличними (Tabular) чи графічними (Graph, BDD) моделями, моделями на системному рівні (System Level). До методів моделювання несправностей можна віднести: апаратні (Hardware), програмні (Software), квантові (Quantum), паралельні (Parallel), дедуктивні (Deductive) та спільні (Concurrent) методи, апаратне вбудоване моделювання (HES), методи активізації критичних шляхів (Critical PA) та парної активізації критичних шляхів (Critical PPA), послідовне моделювання несправностей (Serial FS). Існують наступні моделі несправностей: функціональні (Functional), множинні (Multiple), мостикові (Bridging), константні (Stuck-at), струмкові (IDDQ), несправності типу «переплутування» (Entanglement), несправності, пов'язані з затримками вентиля (Gate Delay F), шляху (Path Delay F), переходу (Transition DF), лінії (Line Delay F) та сегмента (Seg. Delay F), несправності, пов'язані з брязкотом сигналу (Oscillation), модель несправності у семизначному (7v-Alphabet) та шістнадцятизначному (16v-Alphabet) алфавітах моделювання перехідних процесів, а також у дев'ятизначному алфавіті Хейеса (Hayes-Alph.) та у шістнадцятизначному алфавіті Фантози (Fantauzzi Al.) для моделювання перехідних процесів. Всі методи моделювання оперують наступними поняттями (Concepts): специфікація несправнос-

тей (F Specif.), введення несправностей в схему (F Insertion), просування несправностей від входів до виходів (FE Propag.), скидання шляхів активізації несправностей (F discarding).

Основні проблеми в галузі моделювання несправностей слід визначити наступним чином. 1. Висока обчислювальна складність комбінаторних алгоритмів аналізу несправностей на вентильному рівні і рівні регістрових передач опису моделі [2, 5, 8, 11, 14]. 2. Складність алгоритмів моделювання та аналізу результатів моделювання послідовних схем, яка пов'язана з непередбачуваною кількістю ітерацій [1, 2, 6, 7, 10, 13, 14, 15]. 3. Значний обсяг структур даних для аналізу цифрових систем на кристалі, що негативно впливає на продуктивність методів моделювання несправностей і синтезу тестів [1, 3, 5, 9, 18, 20]. 4. Складність аналізу і синтезу тестів для логіки великої розмірності і їх структурна складність, пов'язана з реконвергентними розгалуженнями [1, 4, 8, 9, 11, 18]. 5. Паралельне вирішення задач тестування та моделювання цифрових пристроїв [1, 7, 18 – 30].

3 МАТЕРІАЛИ ТА МЕТОДИ

Розглянемо метрику тестування цифрових систем. Дедукція (deductio) – логічне отримання висновку від загального до конкретного, де правильність гарантується істинністю посилок-аксіом, що призводять до істинності наслідків-теорем. Індукція (inductio) – логічне отримання висновку від часткового до загального, де правильність гарантується достатньою або вичерпною кількістю фактичних даних. Аксіома конволюції testing простору: «Хог-відношення відстаней між кінцевим числом замкнених в циклі компонентів завжди дорівнює нулю» [18]. Очевидність цієї аксіоми демонструється на рис. 2, де циклічне з'єднання кінцевої множини моделей об'єктів завжди створює нульову суму відстаней: $\oplus_{i=1}^n d_i = 0$.

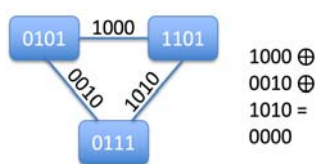


Рисунок 2 – Метрика конволюційного замикання відстаней

Тут складання всіх кодів-відстаней контуру дає результат $1000 \oplus 1010 \oplus 0111 = 0000$. Природно, що аксіома вірна для будь-якого кінцевого числа елементів, що становлять контур або цикл. Слідство: якщо відомі дві відстані, то за хог-відношенням між ними завжди можна визначити третю. Виключною властивістю сукупності взаємопов'язаних компонентів – функція (function), тест (test) і несправність (fault) – є той факт, що одночасно вони являють собою відстані-відносини між усіма парними сполученнями в структурі їх трикутної хог-взаємодії (рис. 3).

Слід мати на увазі, що формати тесту, функції і несправностей повинні бути однієї розмірності. Тому

модель завдання одиночної несправності повинна збігатися з сумарною кількістю входних і вихідних змінних.

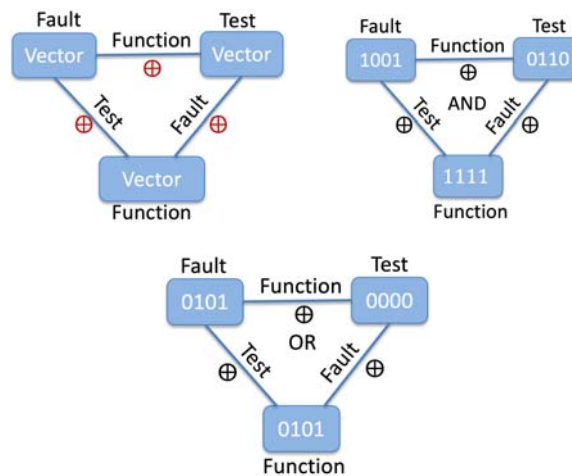


Рисунок 3 – Метрика хог-відносин: function-test-fault (функція-тест-несправність)

Наприклад, для and-or-логіки від трьох змінних несправність першого входу матиме вигляд вектора (1001). Це означає, що несправність першого входу робить істотний вплив на стан виходу, пов'язаного з координатою 4. Test-вектор 0100 (0000), Fault-вектор 1001 і Function-вектор 1111 (0101) для 3and і 3or логіки відповідно представлені в другому і третьому трикутнику. Відносини представлені ребрами трикутної графової структури. При цьому вершини є вектори (таблиці) відносин.

Іншими словами, в трикутній структурі є вершини (вектори-таблиці), які позначають функції-тести-несправності. Однак кожен з цих компонентів створює відносини між двома іншими за допомогою хог-операції. Це означає, що хог-відношення між несправністю (Fault) і тестом (Test) є функцією (Function), між несправністю (Fault) і функцією (Function) є тест-відношення (Test), між тестом (Test) та функцією (Function) є несправність-відношення (Fault) [18].

Трикутна структура таблиць істинності (Function), тестів (Test), несправностей (Fault) для логічного елемента 2and дозволяє вирішувати вже комбінаторні задачі: синтезу тестів, моделювання несправностей і формування таблиці істинності (рис. 4) [18, 28].

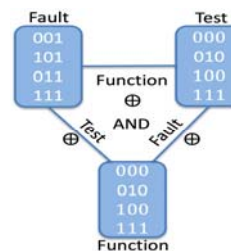


Рисунок 4 – Метрика табличних хог-відносин: function-test-fault (функція-тест-несправність) для 2and-логіки

В цьому випадку трикутна структура відносин між тестом, несправностями та функцією, поданих таблицею істинності, служить шаблоном-стандартом для верифікації отриманих відносин між згаданими знайденими компонентами.

У компоненті Faults одиницями показані несправності (активності ліній), що транспортуються на виході елемента. Перший куб 001 – перевірка несправності виходу, другий куб 101 – активізація першої та вихідної змінної, третій 011 – транспортування несправностей другою змінною на вихід, 111 – транспортування несправностей з двох входів на вихід. Інтерес також представляє схематична форма відносин, яка дає можливість створювати цифрові структури для вирішення задач тестування (рис. 5). Однак, як правило, обчислювальна NP-складність задач тестування [1, 11] пов'язана з наявністю початкових умов – функції (Function) у вигляді: таблиці істинності, рівняння або альтернативних графів.

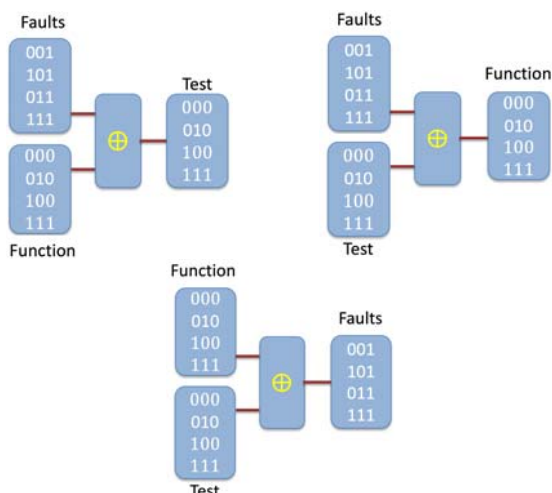


Рисунок 5 – Схематична форма відносин для рівняння тестування

При цьому необхідно знайти тест (Test), а потім його верифікувати на класі одиночних константних несправностей, що означає побудувати два відсутніх компонента в трикутнику рівняння тестування (рис. 6).

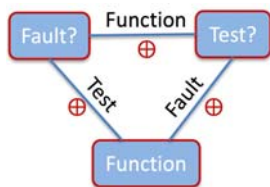


Рисунок 6 – Метрика табличних хог-відносин: function-test-fault (функція-тест-несправність)

Аксиома: «У світі Design and Test немає нічого, крім функцій F , тестів T і несправностей L , хог-відносини яких дорівнюють нулю: $F \oplus T \oplus L = 0$ » [18]. Інша справа, що ця тривіальність і істинність

даної метрики не завжди проглядається через призму складних моделей, методів і алгоритмів проектування, виробництва та експлуатації технічних виробів, напрацьованих людством за останні 100 років. Лінійний характер взаємодії дає можливість визначити будь-який з заданих компонентів через хог-відношення двох атрибутів, що залишилися. Природно, що сукупність несправностей деякої функціональності на тестових наборах можна визначити за допомогою виразу $L = T \oplus F$, яке вироджується в групу відомих методів моделювання несправностей, таких як дедуктивні, паралельні, спільні, одиночні.

Дедуктивне моделювання несправностей можна структурувати на дві послідовні фази.

1. Синтез дедуктивних формул L для логічних або функціональних елементів F комбінаційної схеми [18, 22, 25], що залежать від тестових вхідних наборів T і логіки компонентів $L = f(T, F)$.

2. Моделювання несправностей логічної комбінаційної схеми шляхом використання дедуктивних формул на тестових вхідних послідовностях [27, 30]. Як правило, розглядаються поодинокі константні несправності вхідних, внутрішніх і вихідних ліній схеми, які покривають до 95% всіх можливих несправностей. Сутність дедукції для отримання аналітичних виразів, орієнтованих на транспортування списків несправностей через функціональний або логічний елемент, полягає у використанні хог-операції для формування основного конволюційного лінійного рівняння тестування $T \oplus F \oplus L = 0$. Дана рівність описує всі можливі процеси, пов'язані з синтезом тестів $T = F \oplus L$ і функціональних описів $F = T \oplus L$, діагностуванням несправностей, включаючи визначення вихідного списку несправностей, одержуваного шляхом дедуктивної (теоретико-множинної) взаємодії списків вхідних несправностей, що транспортуються через логічний елемент: $L = T \oplus F$. Тут \oplus – операція виконує глобальну роль вимірника (арбітра) подібності-відмінності між тріадою компонентів: тест, функція, несправності, які повинні бути однаково задані в будь-якому з відомих форматів опису даних: таблиці, графі, рівняння. Таким чином, маючи два відомих компонента в рівнянні тестування, завжди можна отримати будь-який третій шляхом синтезу хог-взаємодії між двома першими. Істинність рівняння тестування можна продемонструвати на простому прикладі логічного інвертора: $F = (01) \oplus T = (10) \oplus L = (11) = (00)$.

Розглянемо таблично-векторні моделі для опису і аналізу структур і функцій. Мета – створення універсальної векторної моделі опису структур і функцій для розробки ефективних паралельних методів синтезу та аналізу, тестування і моделювання компонентів SoC. Поставлені наступні завдання. 1. Обґрунтування кубітно-векторної форми опису структур і функцій. 2. Використання векторної форми для опису графових моделей. 3. Застосування векторної форми для опису таблиць переходів для кінцевих автоматів. 4. Викори-

стання кубітно-векторної форми для розробки векторно-орієнтованих методів синтезу тестів і моделювання несправностей графових структур.

Практичний інтерес представляє пошук простої, технологічної та універсальної форми для опису графів і функцій, алгоритмів і структур, моделей і методів, яка дає можливість застосовувати напрацьовані вченими теорії для ефективного вирішення завдань проектування і тестування, синтезу та аналізу цифрових систем. Розглядаючи аналітичні, табличні і графові моделі [1, 11, 14, 18, 24, 28] можна прийти до таких висновків. 1. Аналітика, володіючи певною компактністю, потребує складних алгоритмів розв'язання рівнянь. 2. Таблиці або матриці мають недолік, пов'язаний з розмірністю моделей для логічних схем від великого числа (> 20) змінних. 3. Графові моделі зручні для візуального сприйняття людиною, але важкі для їх дешифрування і подальшого структурного синтезу цифрового обчислювача. Пропонується серединна векторна (кубітна) форма опису, універсально придатна для завдання функцій і структур, яка має: 1) компактність порівняно з таблицями; 2) простоту опису порівняно з логічними рівняннями; 3) технологічність і паралелізм вирішення завдань аналізу та синтезу; 4) формування нової обчислювальної технології, заснованої на використанні швидкодіючої пам'яті і виключення традиційної логіки зі складу універсального комп'ютера; 5) подальшу орієнтацію алгоритмів і структур даних на квантові обчислювачі. Недоліком і одночасно перевагою векторної форми опису функцій і структур є адресованість його бітів, яка дає можливість швидко (за один цикл звернення до пам'яті) знаходити рішення.

Далі розглядаються дві форми опису: таблиці і вектори, інваріантні для завдання примітивів функцій і структур.

Таблиця істинності $C = (C_1, C_2, \dots, C_i, \dots, C_n)$, де $C_i = \{X_i Y_i\}$, представлена сукупністю вхідних двійкових векторів X_i , де кожному з них ставиться у відповідність стан виходу Y_i .

Інша інтерпретація: таблиця істинності є вектор $Y = (Y_1, Y_2, \dots, Y_i, \dots, Y_n)$, станів виходу логічного елемента, де адресою комірки вектора є двійковий вхідний набір.

Третя інтерпретація: таблиця істинності є орієнтований граф G , представлений повною сукупністю дуг – парами вершин $G = (V_1, V_2, \dots, V_i, \dots, V_n)$, де $V_i = \{V_i^X, V_i^Y\}$, а наявність (відсутність) дуги між

двійково кодovаними вершинами відзначається одиничним (нульовим) значенням стану виходу Y_i .

Четверта інтерпретація: таблиця істинності графа є вектор $Y = (Y_1, Y_2, \dots, Y_i, \dots, Y_n)$, наявності-відсутності дуг, де адресою комірки Y_i вектора є пара двійкових кодів вершин витокy і стокy $\{V_i^X, V_i^Y\}$. При цьому розмірність таблиці істинності $n = 2^k$ завжди має функціональну залежність від числа змінних k . Число двійкових змінних k для таблично-векторного опису графа є функцією від цілої частини логарифма для кількості N вершин графової структури $k = 2 \times \lceil \log_2 N \rceil$, округленої в більший бік. Наприклад, для двох вершин графа (рис. 7) число змінних буде дорівнювати двом $k = 2 \times \lceil \log_2 N \rceil = 2$.

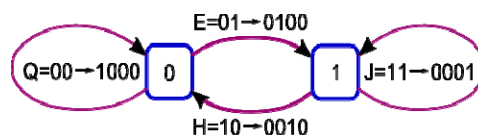


Рисунок 7 – Повний граф з двох вершин і чотирьох дуг

При цьому існує 16 можливих варіантів графових структур, які визначаються комбінаціями вектора Y , поданими у таблиці 1. Тут формула подвоєння числа змінних унеможливує їх непарну кількість, що характерно для відношення, що задається декартовим квадратом вершин графа.

Однак існує специфіка відношення-відповідності, яке формується декартовим добутком окремо кодovаних вершин, що оптимізує число змінних у бік мінімуму і створює непарне число змінних, яке визначається формулою: $k = \lceil \log_2 N_1 \rceil \times \lceil \log_2 N_0 \rceil$, як підмножина декартова добутку двох множин.

Що стосується маркування переходів, характерного для кодування граф-схем алгоритмів або керуючих автоматів, то тут таблиця істинності приростає змінними управління і не більше того.

Недоліком такої форми опису автоматних переходів є розмірність таблиці, яка експоненціально залежить від числа змінних управління U і змінних кодування станів автомата:

$$C = 2 \times \lceil \log_2 N \rceil \lceil +U \times 2^{2 \times \lceil \log_2 N \rceil} \rceil$$

Однак розмірність вектора автоматних переходів є прийнятною для вирішення задач синтезу та аналізу, тестування і діагностування, визначається оцінкою:

$V = 2^{2 \times \lceil \log_2 N \rceil + U}$, яка в $2 \times \lceil \log_2 N \rceil + U$ раз менше таблиці істинності.

Таблиця 1 – Варіанти графових структур

V_i^X	V_i^Y	Y1	Y2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Розглянемо логічні функції вимірювання. Як не дивно, але в природі існують лише дві логічних функції для вимірювання процесів і явищ, що складають групу X -функцій: xor and not-xor [18, 22]. Їм ставляться у відповідність тільки два поняття: відмінності і подібності процесів і явищ, які покладені в основу всіх вимірювальних комп'ютерних технологій, включаючи ML і AI. Метрична унікальність даних функцій визначається їх властивостями немінімізованості (незмінності), що ілюструється наступними топологічними діаграмами Карно їх таблиць істинності від однієї, двох, трьох і чотирьох змінних, які представлені на рис. 8.

X_1	0	1
	0	1

$X_1 \setminus X_2$	0	1
0	0	1
1	1	0

$X_1 \setminus X_2 X_3$	00	01	11	10
0	1	0	1	0
1	0	1	0	1

X_1	0	1
	1	0

$X_1 \setminus X_2$	0	1
0	0	1
1	1	0

$X_1 \setminus X_2 X_3$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Рисунок 8 – Таблиці істинності X -функцій

Таких xor -функцій всього дві від будь-якого кінцевого числа змінних: від однієї змінної (повторювач і інвертор) на тлі загальної кількості, рівної чотирьом; їх всього дві від двох змінних на тлі шістнадцяти (що виключає або, операція еквівалентності); дві від трьох змінних на тлі 256; дві від чотирьох змінних на тлі числа 65536, що визначає їх загальну кількість. Рекурентний синтез X -функцій від довільного числа змінних досить повно представлений в [18, 22]. Слід зауважити, що таблиця істинності, записана в сусідньому кодї, становить певний інтерес для синтезу (тестів), аналізу, мінімізації цифрових схем.

Розглянемо процес вимірювання векторів опису процесів і явищ. Що стосується метрики вимірювання процесів і явищ, представлених двійковими (чисельними) векторами параметрів, то тут фігурують дві оцінки (similarity-difference) [27], які взаємно доповнюють один одного до норми-універсуму, що дорівнює 1. Структурно, метрика подібності-відмінності двох процесів, явищ, об'єктів, компонентів використовує дві формули, які оперують в бінарній алгебрі логіки двома паралельними операціями and , xor для отримання результуючих векторів:

$$(a,b) = a_i \wedge b_i; \quad D(a,b) = a_i \oplus b_i.$$

Нормована метрика подібності-відмінності [16, 17] використовує дві формули, що також оперують в алгебрі логіки двома паралельними операціями, але доповнені арифметикою підрахунку одиничних координат, отриманих в результаті виконання логічних операцій. Крім того, з'являється спільний знаменник у вигляді диз'юнкції однойменних координат векторів, який служить інтегратором розрізних структур да-

них, процесів, що беруть участь, в загальний вектор істотних координат, щодо яких виконується нормування подібності (S) та відмінності (D):

$$S(a,b) = \frac{\sum_{i=1}^n a_i \wedge b_i}{\sum_{i=1}^n a_i \vee b_i}; \quad D(a,b) = \frac{\sum_{i=1}^n a_i \oplus b_i}{\sum_{i=1}^n a_i \vee b_i}.$$

Розглянемо дедуктивне моделювання, а саме синтез та аналіз дедуктивної логіки. Базове рівняння визначається наступною формулою [1, 18]: $L = f(T, F) = T \oplus F$, що випливає з тотожності ($F \oplus L \oplus T = 0$).

Нехай ϵ булева функція від n змінних: $Y = F(x_1, x_2, \dots, x_i, \dots, x_n)$. Згідно з базовим рівнянням, дедуктивна функція може бути зображена у вигляді:

$$F = L \oplus T \rightarrow F(x_1, x_2, \dots, x_i, \dots, x_n) = (X_1 \oplus T_1, X_2 \oplus T_2, \dots, X_i \oplus T_i, \dots, X_n \oplus T_n) \oplus Y_i.$$

Тут $x_i = X_i \oplus T_i$ ($F = L \oplus T$) – визначення вхідної змінної x_i логічної функції через xor -взаємодію списку або вектора вхідних несправностей X_i з тестовим значенням T_i . При цьому $F \oplus Y_i \in \text{xor}$ -взаємодія функції з її тестовим значенням. Якщо, наприклад, $x_i = 0, T_i = 0$ для логічного елемента “and”, то $X_i = 0$, оскільки $x_i \oplus X_i \oplus T_i = 0$. Але якщо $x_i = 0, T_i = 1$, то залишається єдине – $X_i = 1$. Це означає, що тестовий набір, який перевіряє несправність, повинен бути протилежним за значенням до стану вхідної змінної x_i логічної функції. Вірно і те, що трьох одиниць не може бути в компонентах рівності. Всі можливі варіанти довизначення компонентів в тотожність $F \oplus L \oplus T = 0$: всі нулі або дві одиниці з трьох.

Завжди тільки дві функції від кінцевого числа ($n = 1, 2, 3, \dots, k$) змінних (xor , not-xor), які складають сімейство X -функцій [22], мають властивість транспортування несправностей, інваріантну по відношенню до тест-вектору, що робить ці функціональності унікальними.

Таким чином, правило дедукції, саме в вигляді $F = L \oplus T$ для синтезу відповідної логіки, використовує деяку функцію F , яка аксіоматично і парадоксально є відстанню між тестом T і несправностями L . Що стосується функцій від однієї змінної $Y = x_1, Y = \overline{x_1}$, то їх дедуктивний аналіз призводить до тривіального результату, який показує інваріантність транспортування всіх несправностей на вихід щодо будь-якого тестового набору:

$$F = L \oplus T : x_1 = (X_1 \oplus T_{i1}) \oplus Y_i =$$

$$= \overline{x_1}[(X_1 \oplus 0) \oplus 0] \vee x_1[(X_1 \oplus 1) \oplus 1] = \overline{x_1}X_1 \vee x_1X_1 = X_1.$$

$$F = L \oplus T : x_1 = (X_1 \oplus T_{i1}) \oplus Y_i =$$

$$= \overline{x_1}[(\overline{X_1} \oplus 0) \oplus 1] \vee x_1[(\overline{X_1} \oplus 1) \oplus 0] = \overline{x_1}X_1 \vee x_1X_1 = X_1.$$

Отримані дедуктивні вирази не мають вхідних змінних (x_1), що свідчить про їх неістотність для транспортування несправностей (X_1) через елементи: повторювач та інвертор.

Розглянемо векторний (кубітний) метод синтезу дедуктивних функцій у вигляді діз'юнктивної нормальної форми (ДНФ, Disjunctive Normal Form). Метод орієнтований на логіку довільної розмірності, яка задається кубітним вектором функціональності, що робить його практично орієнтованим, з огляду на технологічність і простоту реалізації. Як правило, методи синтезу дедуктивних формул для цифрових схем залишаються за межами публікацій, які описують використання вже готових логічних виразів. Далі пропонується підхід до отримання таких формул, а точніше матриць для дедуктивного аналізу цифрових схем. Найпростішою і ефективною технологією синтезу дедуктивних функцій для моделювання несправностей є метод, який використовує кубітні або векторні покриття булевих функцій. Під кубітним Q -покриттям розуміється вектор станів вихідної змінної таблиці істинності [18, 23, 28], де кожній координаті вектора ставиться у відповідність двійкова адреса, що ототожнюється з вхідним набором функціонального елемента. Якщо врахувати той факт, що для кожного вхідного набору існує власна дедуктивна функція, то для всіх вхідних послідовностей необхідно побудувати матрицю розмірності $2^n \times 2^n$. Процесор синтезу дедуктивної матриці (Matrix of Deductive Q-coverages) для паралельного аналізу одиночних константних несправностей на основі Q -покриття (Q-coverage) функціональності містить тільки дві операції (рис. 9).

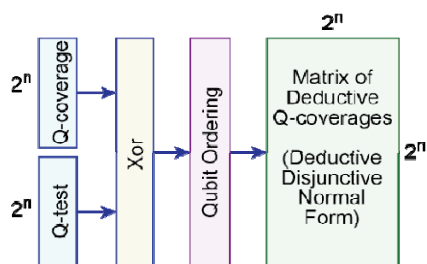


Рисунок 9 – Секвенсор синтезу векторів дедуктивної матриці

1) Синтез матриці Q -векторів дедуктивних функцій залежить від двійково-десятькового номера вхідного набору і координат Q -вектора (Q -test) для функції від n змінних: $L_{ij}^* = (Q_i \oplus Q_j)_{j=1, m}^m$, $m = 2^n$. Фактично береться перша координата кубітного покриття (векто-

ра), яка хог-взаємодіє з усіма координатами Q -вектора функціональності для формування дедуктивного вектора щодо першої вхідної послідовності (Qubit Ordering). Потім береться друга координата Q -вектора, яка також хог-взаємодіє з усіма координатами Q -покриття. Процедура закінчується після того, як всі координати Q -вектора були хог-складені з кубітним вектором. Обчислювальна складність даної процедури дорівнює 2^{n+1} , яка може бути зменшена до n шляхом апаратної реалізації паралельного виконання хог-операції.

2) Після отримання дедуктивної матриці для всіх вхідних наборів по кубітному покриттю функціональності необхідно виконати процедуру перестановки бітів в стовпцях отриманої матриці за правилом: $L_{ij} = (L^*(H_{ij}))_{i,j=1}^m$, відповідно до номерів, поданих в матриці перестановки H , яка за розмірністю дорівнює матриці кубітних покриттів дедуктивних функцій.

Розглянемо кубітно-векторний метод синтезу тестів. Метод становить інтерес завдяки своїй інноваційності, створюваної векторними структурами даних, на яких виконується обчислення похідних за кожною вхідною змінною логічної схеми. Тут похідна визначається виконанням хог-операції над лівою та правою частинами кубітного покриття-вектора [23, 24, 29]:

$$Q'(X_k) = \{Q_i^L, Q_i^R\} = Q_i^L \bigoplus_{i=1, 2^{k-1}}^{k=1, n} Q_i^R.$$

Алгоритм обчислення похідних за змінними для Q -вектора (10001001) логічної схеми:

1. Хог-складання сусідніх бітів Q -вектора з подальшим занесенням результату в кожен із сусідніх бітів:

$$Q'(X_k)(0,1) = Q_0 \oplus Q_1, Q'(X_1)(2,3) = Q_2 \oplus Q_3,$$

$$Q'(X_1)(4,5) = Q_4 \oplus Q_5, Q'(X_1)(6,7) = Q_6 \oplus Q_7.$$

2. Хог-складання сусідніх пар бітів кубітного вектора з подальшим занесенням результату в кожен сусідню пару бітів:

$$Q'(X_2)(0,1,2,3) = Q_{01} \oplus Q_{23},$$

$$Q'(X_2)(4,5,6,7) = Q_{45} \oplus Q_{67}.$$

3. Хог-складання сусідніх тетрад бітів кубітного вектора з подальшим занесенням результату в кожен сусідню тетраду бітів:

$$Q'(X_3)(0,1,2,3,4,5,6,7) = Q_{0123} \oplus Q_{4567}.$$

Результати обчислення похідних представлені таблицею 2.

Таблиця 2 – Похідні за змінними для Q -вектора

Q	1	0	0	0	1	0	0	1
$Q'(X_1)$	1	1	0	0	1	1	1	1
$Q'(X_2)$	1	0	1	0	1	1	1	1
$Q'(X_3)$	0	0	0	1	0	0	0	1

4. Саме наявність векторної структури даних робить ефективним обчислення похідних, що призводить до простої заключної процедури синтезу тестів, яка зводиться до виконання наступної формули $T(Q') = \bigvee_{i=1,n} Q'(X_i)$ для створення векторної структури

ри тесту для такої схеми (рис. 10). Кубітне покриття цифрової схеми представлено вектором станів вихідної змінної (100000000000001) для шістнадцяти вхідних наборів, де були отримані чотири кубітних (векторних) похідних, що становлять тест на основі використання виразу $T(Q') = \bigvee_{i=1,n} Q'(X_i)$.

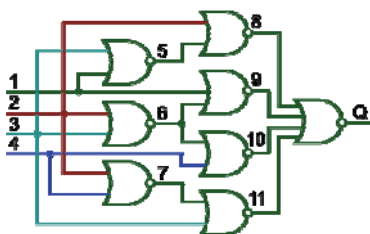


Рисунок 10 – Схема для синтезу тестів

Сукупність кубітних похідних або пар вхідних наборів є тест для перевірки константних несправностей розглянутої вхідної змінної. Таких пар для кожної вхідної лінії може бути кілька, зважаючи на наявність всередині схеми паралельних логічних шляхів від входу до виходу.

4 ЕКСПЕРИМЕНТИ

Розглянемо як приклад відношення, описане таблицею істинності (вектором) і відповідним графом (рис. 11).

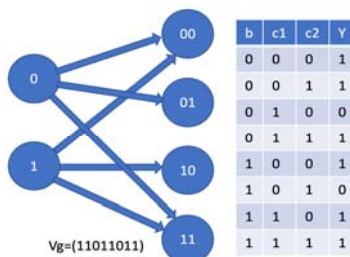


Рисунок 11 – Векторне і табличне подання графа відповідності

Звичайно, даний граф можна представити і в універсальній формі відношення, як підмножини декартова квадрата, що збільшить число змінних для кодування вершин від трьох до шести. Однак така надмірність дає можливість не зберігати інформацію про специфічний розподіл змінних і вершин на дві під-

© Хаханова А. В., Хаханов В. І., Чумаченко С. В., Литвинова Є. І., Рахліс Д. Ю., 2021
 DOI 10.15588/1607-3274-2021-3-7

множини, які формують відповідність. Прикладом автоматного графа може служити таблиця істинності (рис. 12).

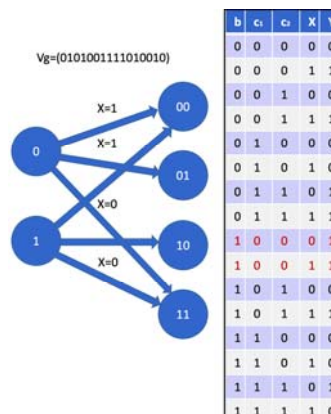


Рисунок 12 – Граф автомата зі змінними для управління переходами

Наступний граф (рис. 13) показує ефективність опису в формі таблиці та вектора.

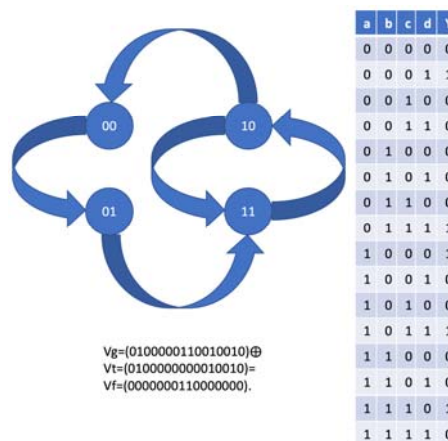


Рисунок 13 – Таблиця істинності і вектор переходів графа

Цей вектор може бути використаний для вирішення завдань тестування, включаючи створення моделі несправностей, синтез тестів для них, моделювання несправностей векторно-дедуктивним методом. Дані технології, орієнтовані на табличну і векторну форму подання функцій і структур, описані публікаціями [18–20]. Наприклад, константні несправності координат вектора Y ідентифікують несправності наявності-відсутності переходів: 1) константа 0 створює помилку відсутності переходу, коли він є в моделі, 2) константа 1 – наявність помилкового переходу, коли його насправді немає. Як приклад тут подані три вектори: опису графа V_g , завдання тесту V_t , а також результат їх взаємодії $V_f = V_g \oplus V_t$ в формі вектора несправностей V_f введеного класу, які перевіряються тестом V_t . Інтерпретація тестування несправностей в векторі $V_f = (0000000110000000)$: перевірка відсутності пе-

реходів між вершинами 01 – 11 та 10 – 00, які повинні бути в специфікованій моделі автомата. На рис. 14 зображено граф переходів \overline{RS} -тригера, який має три стійких стани 01, 10, 11, а також один неможливий – 00.

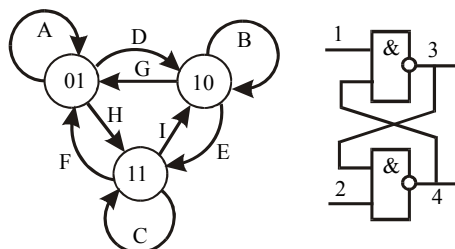


Рисунок 14 – Граф переходів \overline{RS} -тригера

Наступна таблиця істинності (табл. 3) являє собою проміжні дані для формування векторної форми опису графа переходів, яка записана в правій колонці $Y = (0000011101\ 110111)$.

Таблиця 3 – Таблиця істинності

X1	X2	Y1	Y2	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Досить просто визначити потужність несправностей переходів, які прив'язуються до повного графу, що становить $2 * n$ можливих несправностей, протилежних за знаком вектору стану таблиці істинності $Y = (111110001\ 0001000)$. Природно, що синтезований тест повинен покривати згенеровану множину несправностей.

Наприклад, два вектора $a = 00111100$ і $b = 10101010$, що мають несуттєві нульові однойменні координати, автоматично виключаються з нормованого оцінювання, завдяки врахуванню і підрахунку тільки одиничних значень в результуючих векторах:

$$S(a,b) = \frac{\sum_{i=1}^n (00111100 \wedge 10101010 = 001010000) = 2}{\sum_{i=1}^n (00111100 \vee 10101010 = 10111110) = 6} = 0,33;$$

$$D(a,b) = \frac{\sum_{i=1}^n (00111100 \oplus 10101010 = 10010110) = 4}{\sum_{i=1}^n (00111100 \vee 10101010 = 10111110) = 6} = 0,66.$$

Розглянемо синтез дедуктивної логіки для and-елемента:

$$Y = x_1 \wedge x_2 = x_1 x_2 \rightarrow (X_1 \oplus T_1)(X_2 \oplus T_2) \oplus Y.$$

$$L[T = (00,01,10,11), F = (x_1 \wedge x_2)] = (00 \vee 01 \vee 10 \vee 11) \wedge [(X_1 \oplus T_{i1})(X_2 \oplus T_{i2}) \oplus Y_i] = L\{\overline{x_1 x_2} \vee \overline{x_1} x_2 \vee x_1 \overline{x_2} \vee x_1 x_2\} [(X_1 \oplus T_{i1})(X_2 \oplus T_{i2}) \oplus Y_i] = \overline{x_1 x_2} [(X_1 \oplus 0) \wedge (X_2 \oplus 0) \oplus 0] \vee \overline{x_1} x_2 [(X_1 \oplus 0)(X_2 \oplus 1) \oplus 0] \vee x_1 \overline{x_2} \wedge [(X_1 \oplus 1)(X_2 \oplus 0) \oplus 1] \vee x_1 x_2 [(X_1 \oplus 1)(X_2 \oplus 1) \oplus 1] =$$

$$= \overline{x_1 x_2} (X_1 X_2) \vee \overline{x_1} x_2 (X_1 \overline{X_2}) \vee x_1 \overline{x_2} (\overline{X_1} X_2) \vee x_1 x_2 (X_1 X_2).$$

Цікаво, що синтез дедуктивної логіки для xor-елемента створює функцію, рівну вихідній:

$$Y = x_1 \oplus x_2 \Rightarrow (X_1 \oplus T_1) \oplus (X_2 \oplus T_2) \oplus Y.$$

$$L[T = (00,01,10,11), F = (x_1 \oplus x_2)] = (00 \vee 01 \vee 10 \vee 11) [(X_1 \oplus T_{i1}) \oplus (X_2 \oplus T_{i2}) \oplus Y_i] = L\{\overline{x_1 x_2} \vee \overline{x_1} x_2 \vee x_1 \overline{x_2} \vee x_1 x_2\} [(X_1 \oplus T_{i1}) \oplus (X_2 \oplus T_{i2}) \oplus Y_i] =$$

$$= \overline{x_1 x_2} [(X_1 \oplus 0) \oplus (X_2 \oplus 0) \oplus 0] \vee \overline{x_1} x_2 \wedge \wedge [(X_1 \oplus 0) \oplus (X_2 \oplus 1) \oplus 0] \vee x_1 \overline{x_2} [(X_1 \oplus 1) \oplus (X_2 \oplus 0) \oplus 1] \vee x_1 x_2 [(X_1 \oplus 1) \oplus (X_2 \oplus 1) \oplus 1] =$$

$$= \overline{x_1 x_2} (X_1 \oplus X_2) \vee \overline{x_1} x_2 (X_1 \oplus X_2) \vee x_1 \overline{x_2} (X_1 \oplus X_2) \vee x_1 x_2 (X_1 \oplus X_2) = X_1 \oplus X_2 = \overline{X_1} X_2 \vee X_1 \overline{X_2}.$$

Аналогічний результат має ще тільки одна функція – not-xor:

$$Y = \overline{x_1 \oplus x_2} \Rightarrow \overline{(X_1 \oplus T_1) \oplus (X_2 \oplus T_2)} \oplus Y.$$

$$L[T = (00,01,10,11), F = \overline{x_1 \oplus x_2}] = (00 \vee 01 \vee 10 \vee 11) \wedge \wedge [(X_1 \oplus T_1) \oplus (X_2 \oplus T_2) \oplus Y_i] = L\{\overline{x_1 x_2} \vee \overline{x_1} x_2 \vee x_1 \overline{x_2} \vee x_1 x_2\} [(X_1 \oplus T_1) \oplus (X_2 \oplus T_2) \oplus Y_i] =$$

$$= \overline{x_1 x_2} [(X_1 \oplus 0) \oplus (X_2 \oplus 0) \oplus 1] \vee \vee \overline{x_1} x_2 [(X_1 \oplus 0) \oplus (X_2 \oplus 1) \oplus 0] \vee \vee x_1 \overline{x_2} [(X_1 \oplus 1) \oplus (X_2 \oplus 0) \oplus 0] \vee \vee x_1 x_2 [(X_1 \oplus 1) \oplus (X_2 \oplus 1) \oplus 1] =$$

$$= \overline{x_1 x_2} (X_1 \oplus X_2) \vee \overline{x_1} x_2 (X_1 \oplus X_2) \vee x_1 \overline{x_2} (X_1 \oplus X_2) \vee x_1 x_2 (X_1 \oplus X_2) =$$

$$= X_1 \oplus X_2 = \overline{X_1} X_2 \vee X_1 \overline{X_2}.$$

Синтез дедуктивної логіки L на тестових наборах T для двухвходового og-елемента F також здійснюється за правилом $F = L \oplus T$: $x_1 \vee x_2 = (X_1 \oplus T_{i1}) \vee (X_2 \oplus T_{i2}) \oplus Y_i$.

Подальші найпростіші перетворення призводять до синтезу дедуктивної логіки:

$$L[T = (00,01,10,11), F = (x_1 \vee x_2)] = (00 \vee 01 \vee 10 \vee 11) \wedge \wedge [(X_1 \oplus T_{i1}) \vee (X_2 \oplus T_{i2}) \oplus Y_i] = L\{\overline{x_1 x_2} \vee \overline{x_1 x_2} \vee x_1 x_2 \vee x_1 x_2\}[(X_1 \oplus T_{i1} \vee X_2 \oplus T_{i2}) \oplus Y_i] =$$

$$= \overline{x_1 x_2} \wedge [(X_1 \oplus 0) \vee (X_2 \oplus 0) \oplus 0] \vee \vee \overline{x_1 x_2} [(X_1 \oplus 0) \vee (X_2 \oplus 1) \oplus 1] \vee \vee x_1 x_2 [(X_1 \oplus 1) \vee (X_2 \oplus 0) \oplus 1] \vee \vee x_1 x_2 [(X_1 \oplus 1) \vee (X_2 \oplus 1) \oplus 1] =$$

$$= \overline{x_1 x_2} (X_1 \vee X_2) \vee \overline{x_1 x_2} (\overline{X_1 X_2}) \vee x_1 x_2 (\overline{X_1 X_2}) \vee x_1 x_2 (X_1 X_2).$$

Далі наведені приклади отримання матриці дедуктивних векторів і відповідних їм функцій для логіки (and, or, xor, not-xor, not, repeater) на основі використання тільки кубітних покриттів функціональностей [18, 24, 27]:

Q _{and}	L ₁	L ₂	L ₃	L ₄
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
1	1	1	1	0

 \rightarrow

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

 \rightarrow

X ₁ X ₂	L ₁	L ₂	L ₃	L ₄
00	0	0	0	0
01	0	0	1	1
10	0	1	0	1
11	1	0	0	1

 $\rightarrow L = (00)(X_1 X_2) \vee (01)(\overline{X_1 X_2}) \vee (10)(\overline{X_1 X_2}) \vee (11)(X_1 X_2).$

Q _{or}	L ₁	L ₂	L ₃	L ₄
0	0	1	1	1
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0

 \rightarrow

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

 \rightarrow

X ₁ X ₂	L ₁	L ₂	L ₃	L ₄
00	0	0	0	0
01	1	1	0	0
10	1	0	1	0
11	1	0	0	1

 $\rightarrow L = (00)(X_1 \vee X_2) \vee (01)(\overline{X_1 X_2}) \vee (10)(\overline{X_1 X_2}) \vee (11)(X_1 X_2).$

Q _{xor}	L ₁	L ₂	L ₃	L ₄
0	0	1	1	0
1	1	0	0	1
1	1	0	0	1
0	0	1	1	0

 \rightarrow

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

 \rightarrow

X ₁ X ₂	L ₁	L ₂	L ₃	L ₄
00	0	0	0	0
01	1	1	1	1
10	1	1	1	1
11	0	0	0	0

 $\rightarrow L = (00 \vee 01 \vee 10 \vee 11)(\overline{X_1 X_2} \vee X_1 \overline{X_2}) = (xx)(\overline{X_1 X_2} \vee X_1 \overline{X_2}).$

Q _{xnor}	L ₁	L ₂	L ₃	L ₄
1	0	1	1	0
0	1	0	0	1
0	1	0	0	1
1	0	1	1	0

 \rightarrow

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

 \rightarrow

X ₁ X ₂	L ₁	L ₂	L ₃	L ₄
00	0	0	0	0
01	1	1	1	1
10	1	1	1	1
11	0	0	0	0

 $\rightarrow L = (00 \vee 01 \vee 10 \vee 11)(\overline{X_1 X_2} \vee X_1 \overline{X_2}) = (xx)(\overline{X_1 X_2} \vee X_1 \overline{X_2}).$

Q _{not}	L ₁	L ₂
1	0	1
0	1	0

 \rightarrow

1	2
2	1

 \rightarrow

X	L ₁	L ₂
0	0	0
1	1	1

 \rightarrow

$$\rightarrow L = (0 \vee 1)(X \vee X) = (xx)(X).$$

Q _{rep}	L ₁	L ₂
0	0	1
1	1	0

 \rightarrow

1	2
2	1

 \rightarrow

X	L ₁	L ₂
0	0	0
1	1	1

 \rightarrow

$$\rightarrow L = (0 \vee 1)(X \vee X) = (xx)(X).$$

Тут процедура перестановки бітів є рекурсивною і регулярною, але не є обчислювально складною [28, 29].

У таблиці 4 наведені: стани вхідних змінних, які відповідають значенням координат кубітного вектора-покриття, булеві похідні, отриманий тест об'єднання похідних (Test) і мінімальний тест-вектор (T_{min}).

Таблиця 4 – Таблиця даних

X ₁11111111
X ₂1111....1111
X ₃	..11..11..11..11
X ₄	.1.1.1.1.1.1.1.1
Q	1.....1
Q'(X ₁)	1.....11.....1
Q'(X ₂)	1...1.....1...1
Q'(X ₃)	1.1.....1.1
Q'(X ₄)	11.....11
Test =	111.1...11...1.111
T _{min} =	11.....11.....11

Для візуального сприйняття таблиці даних точками відзначені нульові стани координат таблиці істинності і матриці похідних. В результаті виконання og-операції над векторами кубітних похідних було отримано тест, який містить 10 вхідних наборів T = (1110100110010111). Він є тестом діагностування для одиночних константних несправностей.

Подальша процедура мінімізації тесту здійснюється на основі дедуктивного моделювання, з урахуванням структури схеми, шляхом знаходження квазімінімального покриття (тут не розглядається) всіх змінних-стовпців таблиці нульовими і одиничними значеннями виявлених константних несправностей.

Виконання даної процедури формує результат з шести тестових наборів, наведених в таблиці 5.

Тут точками відзначені координати з неперевірюваними несправностями, FD – Fault Detected, FC – Fault Coverage, X = {0,1}.

Таблиця 5 – Тестові набори

Test	1	2	3	4	5	6	7	8	9	10	11	12	FD	FC
000011100001	1	1	1	1	0	0	0	1	1	1	1	0	50	50
000111000010	.	.	.	0	.	.	1	.	.	0	1	16	66	
011100001000	1	.	.	0	.	.	1	12	75	
100001110000	0	.	.	1	.	0	.	.	.	1	16	87		
111000000100	1	.	.	.	0	.	1	12	91	
111100000001	0	0	0	0	.	.	.	1	1	1	1	0	37	100
U =	X	X	X	X	X	X	X	X	X	X	X	X		

6 РЕЗУЛЬТАТИ

Таким чином, векторна форма опису дає можливість застосувати напрацьовані вченими технології синтезу та аналізу тестів логічних схем для ефективного вирішення завдань тестування графових структур і автоматних моделей цифрових пристроїв.

При вимірі векторів опису процесів немає потреби обчислювати обидві оцінки за даними формулами. Досить визначити одну з них, а другу можна отримати за формулою доповнення: $D(a, b) = 1 - S(a, b)$; $S(a, b) = 1 - D(a, b)$.

Результати синтезу дедуктивної логіки для базисних елементів (and, or, not) орієнтовані на апаратну реалізацію методу моделювання несправностей, де фігурують вхідні тестові набори в форматі $(x_1 x_2)$, а також вектори або списки вхідних несправностей $(X_1 X_2)$, що транспортуються на вихід логічного елемента.

Запропонований метод синтезу дедуктивних матриць для моделювання несправностей відрізняється від відомих аналогів оригінальністю математичних рішень, високим рівнем паралелізму і компактністю структур даних, що дозволяє використовувати його програмну чи апаратну реалізацію для синтезу, аналізу, тестування, верифікації та діагностування цифрових систем на кристалах. Для розвитку memory-driven logic-free комп'ютерингу запропонований кубітно-векторний дедуктивний аналіз є істотним внеском в проектування ремонтпридатних цифрових систем регістрового і системного рівня опису [18].

Проведені експерименти з синтезу тестів на 16 схемах, що містять 4–10 вхідних змінних, показали: 1) для чотирьох схем були отримані оптимальні тести; 2) для дванадцяти схем тести відрізнялися від оптимальних не більше, ніж на 25 відсотків; 3) обчислювальна складність запропонованого кубітного методу синтезу квазіоптимального тесту для логічної функціональності від n змінних визначається оцінкою, яка формує часові витрати на взяття кубітних похідних і мінімізацію тесту: $Q = 2n + 2n^2 = 2n(n + 1)$.

7 ОБГОВОРЕННЯ

Основне рівняння тестування, яке використовується для цифрових систем, також толерантно і може бути застосовано для тестування графових структур. Природно, вектор графа може бути також перетворений до матричної форми, яка визначає підмножину декартового квадрата щодо кодових позначень вершин. Але в цьому випадку необхідно використовувати вже інші специфічні матричні методи синтезу та аналізу графових структур.

Відмінністю формованої оцінки від відстані Хеммінга є виключення з метрики і структур даних умови існування двох нулів на однойменних координатах, що істотно підвищує адекватність вимірювання процесів.

Інтеграція отриманої дедуктивної логіки (рис. 15) представлена універсальним дедуктивним процесором [18, 26, 27] для обробки логічних схем в базисі (and, or, not), де V – змінна вибору типу логічного елемента для його дедуктивного моделювання з метою транспортування вхідних списків несправностей на вихід.

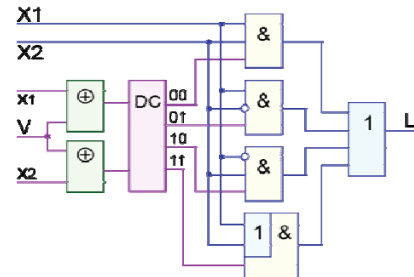


Рисунок 15 – Дедуктивний процесор

У схемі інвертор відсутній через його неістотність (не вплив) на процес транспортування несправностей. Процесор орієнтований на його імплементацію як засобу BIST [1,11], яке може бути використане в режимі онлайн тестування, моделювання та діагностування несправностей для цифрових систем на кристалах вентиляційного рівня. Слід зауважити, що тема мінімізації дедуктивної логіки для схем великої розмірності відсутня в сучасних публікаціях.

Висока продуктивність синтезу дедуктивних кубітних покриттів $Q = 2 \times 2^n = 2^{n+1}$ є підставою для його використання в режимі online, оскільки регістрова реалізація фактично двох операцій: xor-порівняння і перестановки бітів, дозволить звести згадану оцінку до 2–5 автоматних тактів. Крім того, дві згадані операції можна об'єднати в одну процедуру, виконувану в одному автоматному такті. Примітно, що для X -функціональності довільної складності процедура синтезу тестів відсутня, а його запис визначається всіма одиничними координатами Q -вектора, доповненими нульовою координатою. Тест для X -функцій (3-xor, 3-not-xor) формується таким чином:

$$T = T^1 \vee T_i^0;$$

$$T^1 = \forall T_i : f(T_i) = 1;$$

$$T_i^0 \in T^0 = \forall T_i : f(T_i) = 0;$$

$$T(01101001) = (001 \vee 010 \vee 100 \vee 111) \vee 000;$$

$$T(10010110) = (000 \vee 011 \vee 101 \vee 110) \vee 001.$$

Потужність тесту для X -функції від n змінних завжди дорівнює половині довжини кубітного покриття плюс одиниця: $Q = 1 + \frac{1}{2} \times 2^{2^n}$.

ВИСНОВКИ

В роботі розглянута задача, пов'язана з розробкою інноваційної технології кубітно-векторного синтезу і

дедуктивного аналізу тестів для їх верифікації на основі векторних структур даних, що істотно спрощують алгоритми, які можуть бути вбудовані як компоненти BIST в цифрові системи на кристалах.

Наукова новизна отриманих результатів полягає в тому, що:

1) вперше розроблено векторну форму опису структур, яка дає можливість застосувати напрацьовані вченими технології синтезу та аналізу тестів логічних схем для ефективного вирішення задач тестування графових структур та автоматних моделей цифрових пристроїв;

2) вперше отримано нову технологічну метрику хог-відношення в довільному форматі даних: функція-тест-несправність, яка є утвореною для формулювання всіх завдань, пов'язаних з синтезом тестів, функціональних описів і моделюванням несправностей SoC;

3) вперше розроблено новий метод синтезу дедуктивних матриць для моделювання несправностей цифрових схем логічного і RT-рівня, який відрізняється від відомих аналогів оригінальністю математичних рішень, високим рівнем паралелізму і компактністю кубітно-векторних структур даних, що дозволяє використовувати його програмну чи апаратну реалізацію для синтезу, аналізу, тестування, верифікації та діагностування SoC;

4) вперше розроблено новий дедуктивний процесор для моделювання несправностей, який орієнтований на імплементацію як засобу BIST, використовуваного в режимі онлайн тестування, моделювання та діагностування несправностей для цифрових систем на кристалах;

5) вперше отримано нову кубітно-векторну форму опису цифрових систем, яка перевершує існуючі способи завдання обчислювальних пристроїв за метриками: технологічність, компактність, швидкодія і якість. Кубітно-векторне покриття є memo-driven описом функціональності для більш простого рішення задач аналізу, синтезу, тестування і моделювання цифрових пристроїв.

Практична цінність одержаних результатів визначається розробкою програмного додатка QuaSim Services (рис. 16), що реалізує основні сервіси тестування (Test Synthesis), моделювання (Fault Simulation) та діагностування (Fault Diagnosis), що використовуються в навчальному процесі для вивчення переваг кубітно-векторних структур даних (Qubit Data Structures) і алгоритмів (Algorithms) [18, 26, 29]. Бібліотека (Library) включає схеми (Circuits), логічні елементи (Elements), тести (Tests), в тому числі таблицю виявлення несправностей (Fault Detection Table, FDT).

Сервіси мають ергономічний візуальний графічний інтерфейс (GUI "Qubit") [18, 19, 20] для ручного синтезу цифрових логічних схем, де кожен універсальний елемент визначається кубітним покриттям або його десятковим еквівалентом, що істотно зменшує час для введення схемного опису і подальшого дослідження: синтез тестів, діагностування, моделювання несправностей та справної поведінки (modeling and simulation).

© Хаханова А. В., Хаханов В. І., Чумаченко С. В., Литвинова Є. І., Рахліс Д. Ю., 2021
DOI 10.15588/1607-3274-2021-3-7

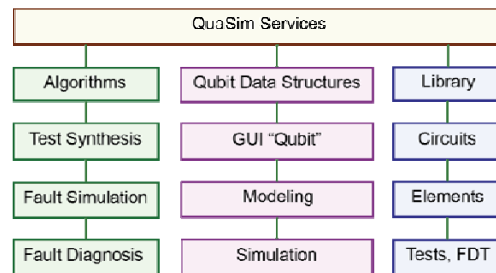


Рисунок 16 – Сервіси тестування на основі векторних структур даних

Перспективи подальших досліджень полягають у імплементації векторних кубітних структур даних в технології класичних обчислювальних процесів, а також подальшу орієнтацію алгоритмів і структур даних на квантові обчислювачі.

ПОДЯКИ

У даній роботі використані результати, отримані авторами у 2019–2021 роках на кафедрі Автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки. Автори вдячні колегам за їх активну участь у обговоренні та підтримці наукової школи «Проектування і технічна діагностика цифрових систем на кристалах, комп'ютерів і мереж», розуміння важливого значення розвитку фундаментальних і прикладних досліджень теорії квантових обчислювальних процесів для аналізу кіберпростору. Всі автори заявляють, що не мають фінансової підтримки.

ЛІТЕРАТУРА / LITERATURA

1. Abramovici M. Digital System Testing and Testable Design / M. Abramovici, M. A. Breuer, A. D. Friedman. – New York : Computer Society Press, 1990. – 652 p.
2. Comparison of Parallel and Deductive Fault Simulation Methods / [H. Y. Chang, S. G. Chappell, C. H. Elmendorf et al.] // IEEE Transactions on Computers. – November 1974. – Vol. C-23, № 11. – P. 1132–1138. DOI: 10.1109/TC.1974.223820.
3. Takahashi N. Fault simulation for multiple faults by Boolean function manipulation / N. Takahashi, N. Ishiura, S. Yajima // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – April 1994. – Vol. 13, № 4. – P. 531–535. DOI: 10.1109/43.275363.
4. Menon P.R. Deductive Fault Simulation with Functional Blocks / P. R. Menon, S. G. Chappell // IEEE Transactions on Computers. – Aug. 1978. – Vol. C-27, № 8. – P. 689–695. DOI: 10.1109/TC.1978.1675175.
5. Fujiwara H. Logic Testing and Design for Testability / H. Fujiwara // Cambridge: MIT Press. – 1985. – P. 84–108.
6. An Efficient Degraded Deductive Fault Simulator for Small-Delay Defects / [T. Liu, T. Yu, S. Wang et al.] // IEEE Access. – 2020. – Vol. 8. – P. 204855–204862. DOI: 10.1109/ACCESS.2020.3037292.
7. Pomeranz I. Forward-looking fault simulation for improved static compaction / I. Pomeranz, S. M. Reddy // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – Oct. 2001. – Vol. 20, № 10. – P. 1262–1266. DOI: 10.1109/43.952743.

8. Kinoshita Y. A Test Generation Method Based on k-Cycle Testing for Finite State Machines / Y. Kinoshita, T. Hosokawa, H. Fujiwara // IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS), Rhodes, Greece, 1–3 July 2019: proceedings. – Rhodes: IEEE, 2019. – P. 232–235. DOI: 10.1109/IOLTS.2019.8854426.
9. Fault modeling and test algorithm creation strategy for FinFET-based memories / [G. Harutyunyan, G. Tshagharyan, V. Vardanian et al.] // IEEE 32nd VLSI Test Symposium (VTS), Napa, CA, 13–17 April 2014: proceedings. – Napa: IEEE, 2014. – P. 1–6. DOI: 10.1109/VTS.2014.6818747.
10. Zolfy M. Adaptation of an event-driven simulation environment to sequentially propagated concurrent fault simulation / M. Zolfy, S. Mirkhani, Z Navabi // Design, Automation and Test in Europe. Conference and Exhibition, Munich, Germany, 13–16 March 2001: proceedings. – Munich : IEEE, 2001. – P. 1530–1591. DOI: 10.1109/DATE.2001.915173.
11. Zainalabedin N. Digital System Test and Testable Design using HDL models and architectures / N. Zainalabedin. – New York: Springer, 2011. – 391 p.
12. Pomeranz I. Unspecified Transition Faults: A Transition Fault Model for At-Speed Fault Simulation and Test Generation / I. Pomeranz, S. M. Reddy // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – Jan. 2008. – Vol. 27, № 1. – P. 137–146. DOI: 10.1109/TCAD.2007.907000.
13. Ubar R. Fast Fault Simulation for Extended Class of Faults in Scan Path Circuits / [R. Ubar, S. Devadze, J. Raik et al.] // Fifth IEEE International Symposium on Electronic Design, Test & Applications, Ho Chi Minh City, Vietnam, 13–15 Jan. 2010: proceedings. – Vietnam: IEEE, 2010. – P. 14–19. DOI: 10.1109/DELTA.2010.32.
14. Fast RTL Fault Simulation Using Decision Diagrams and Bitwise Set Operations / [U. Reinsalu, J. Raik, R. Ubar et al.] // IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Vancouver, BC, 3–5 Oct. 2011: proceedings. – Vancouver: IEEE, 2001. – P. 164–170. DOI: 10.1109/DFT.2011.42.
15. Dobai R. Deductive Fault Simulation for Asynchronous Sequential Circuits / R. Dobai, E. Gramatova // 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools, Patras, 27–29 Aug. 2009 : proceedings. – Patras : IEEE, 2009. – P. 459–464. DOI: 10.1109/DSD.2009.129.
16. Temma S. The Document Similarity Index based on the Jaccard Distance for Mail Filtering / S. Temma, M. Sugii, H. Matsuno // 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), JeJu, Korea (South), 23–26 June 2019: proceedings. – JeJu : IEEE, 2019. – P. 1–4. DOI: 10.1109/ITC-CSCC.2019.8793419.
17. Jaccard P. Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines / P. Jaccard // Bulletin de la Societe Vaudoise des Sciences. Naturelles. – 1901. – Vol. 37, № 140. – P. 241–272. DOI: 10.5169/seals-266440.
18. Hahanov V. Cyber Physical Computing for IoT-driven Services / V. Hahanov. – New York : Springer, 2018. – 279 p.
19. Deductive qubit fault simulation / [I. Hahanov, S. Chumachenko, I. Iemelianov et al.] // 14th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Lviv, 21–25 Feb. 2017: proceedings. – Lviv: IEEE, 2017. – P. 256–259. DOI: 10.1109/CADSM.2017.7916129.
20. Quantum sequencer for the minimal test synthesis of black-box functionality / [V. Hahanov, I. Iemelianov, S. Chumachenko et al.] // IEEE East-West Design & Test Symposium (EWDTS), Novi Sad, 29 Sept.–2 Oct. 2017: proceedings. – Novi Sad: IEEE, 2017. – P. 1–6. DOI: 10.1109/EWDTS.2017.8110148.
21. “Quantum” diagnosis and simulation of SoC / [V. Hahanov, I. Yemelyanov, V. Obrizan et al.] // XI International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, 2–6 Sept. 2015: proceedings. – Lviv : IEEE, 2015. – P. 58–60.
22. Test Synthesis for Logical X-functions / [V. Hahanov, M. Liubarskyi, W. Gharibi et al.] // IEEE East-West Design & Test Symposium (EWDTS), Kazan, 14–17 Sept. 2018: proceedings. – Kazan: IEEE, 2018. – P. 1–9. DOI: 10.1109/EWDTS.2018.8524863.
23. Qubit Test Synthesis Processor for SoC Logic / [W. Gharibi, D. Devadze, V. Hahanov et al.] // IEEE East-West Design & Test Symposium (EWDTS), Batumi, Georgia, 13–16 Sept. 2019: proceedings. – Batumi: IEEE, 2019. – P. 1–5. DOI: 10.1109/EWDTS.2019.8884476.
24. Quantum Models and Method for Analysis and Testing Computing Systems / [V. I. Hahanov, S. M. Hyduke, W. Gharibi et al.] // 11th International Conference on Information Technology: New Generations, Las Vegas, NV, 7–9 April 2014: proceedings. – Las Vegas : IEEE, 2014. – P. 430–434. DOI: 10.1109/ITNG.2014.125.
25. Qubit Fault Detection in SoC Logic / [M. Karavay, V. Hahanov, E. Litvinova et al.] // IEEE East-West Design & Test Symposium (EWDTS), Batumi, Georgia, 13–16 Sept. 2019: proceedings. – Batumi: IEEE, 2019. – P. 1–7. DOI: 10.1109/EWDTS.2019.8884475.
26. Quantum Deductive Simulation for Logic Functions / [V. Hahanov, A. V. Hacimahmud, E. Litvinova et al.] // IEEE East-West Design & Test Symposium (EWDTS), Kazan, 14–17 Sept. 2018: proceedings. – Kazan: IEEE, 2018. – P. 1–7. DOI: 10.1109/EWDTS.2018.8524619.
27. Similarity-Difference Analysis and Matrix Fault Diagnosis of SoC-components / [V. Hahanov, M. Karavay, V. Sergienko et al.] // IEEE East-West Design & Test Symposium (EWDTS), Varna, Bulgaria, 4–7 Sept. 2020: proceedings. – Varna: IEEE, 2020. – P. 1–5. DOI: 10.1109/EWDTS50664.2020.9224740.
28. Qubit-driven Fault Simulation / [V. Hahanov, W. Gharibi, E. Litvinova et al.] // IEEE Latin American Test Symposium (LATS), Santiago, Chile, 11–13 March 2019: proceedings. – Santiago: IEEE, 2019. – P. 1–7. DOI: 10.1109/LATW.2019.8704583.
29. Qubit test synthesis of the functionality / [V. Hahanov, T. B. Amer, E. Litvinova et al.] // 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Lviv, 21–25 Feb. 2017: proceedings. – Lviv: IEEE, 2017. – P. 251–255. DOI: 10.1109/CADSM.2017.7916128.
30. Deductive qubit fault simulation / [I. Hahanov, S. Chumachenko, I. Iemelianov et al.] // 14th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Lviv, 21–25 Feb. 2017: proceedings. – Lviv: IEEE, 2017. – P. 256–259. DOI: 10.1109/CADSM.2017.7916129.

Received 22.03.2021.
Accepted 16.08.2021

ВЕКТОРНЫЕ МОДЕЛИ ЛОГИКИ И СТРУКТУРЫ ДЛЯ ТЕСТИРОВАНИЯ И МОДЕЛИРОВАНИЯ ЦИФРОВЫХ СХЕМ

Хаханова А. В. – канд. техн. наук, профессор каф. Автоматизации проектирования вычислительной техники, Харьковский национальный университет радиоэлектроники, Украина.

Хаханов В. И. – д-р техн. наук, профессор каф. Автоматизации проектирования вычислительной техники, Харьковский национальный университет радиоэлектроники, Украина.

Чумаченко С. В. – д-р техн. наук, профессор, зав. каф. Автоматизации проектирования вычислительной техники, Харьковский национальный университет радиоэлектроники, Украина.

Литвинова Е. И. – д-р техн. наук, профессор каф. Автоматизации проектирования вычислительной техники, Харьковский национальный университет радиоэлектроники, Украина.

Рахлис Д. Е. – канд. техн. наук, доцент каф. Автоматизации проектирования вычислительной техники, Харьковский национальный университет радиоэлектроники, Украина.

АННОТАЦИЯ

Актуальность. Известно, что структуры данных являются определяющими для создания эффективных параллельных алгоритмов и высокопроизводительных вычислительных устройств. Поэтому разработка математически совершенных и технологически простых структур данных занимает около 80 процентов времени проектирования, когда на алгоритмы и их hardware-software кодирование затрачивается около 20 процентов временных и материальных ресурсов. Это обуславливает поиск таких примитивов структур данных, которые существенно упростят работающие на них параллельные высокопроизводительные алгоритмы. Предлагаются модели и методы для тестирования и моделирования цифровых систем, содержащие отдельные преимущества квантового компьютеринга в части имплементации векторных кубитных структур данных в технологии классических вычислительных процессов.

Цель. Разработка инновационной технологии кубитно-векторного синтеза и дедуктивного анализа тестов для их верификации на основе векторных структур данных, существенно упрощающих алгоритмы, которые могут быть встроены в качестве компонентов BIST в цифровые системы на кристаллах.

Метод. Используется дедуктивное моделирование неисправностей для получения аналитических выражений, ориентированных на транспортирование списков неисправностей через функциональный или логический элемент на основе хог-операции, которая выполняет роль измерителя сходства-различия между тестом, функцией и неисправностями, заданными одинаково в одном из форматов – таблицей, графом, уравнением. Предлагается двоичный вектор, как самый технологичный примитив структур данных для задания логической функциональности в целях параллельного синтеза и анализа цифровых систем. Параллелизм решения комбинаторных задач есть физическое свойство квантового компьютеринга, которое в классическом компьютеринге, для параллельного моделирования и диагностирования неисправностей, обеспечивается унитарно-кодированными структурами данных, благодаря избыточной памяти.

Результаты. 1) Разработан метод аналитического синтеза дедуктивной логики для функциональных элементов вентильного уровня и уровня регистровых передач. 2) Предложен дедуктивный процессор для моделирования неисправностей на основе транспортирования входных списков или векторов неисправностей на внешние выходы цифровых схем. 3) Описаны кубитно-векторная форма задания логики и методы кубитного синтеза дедуктивных уравнений для моделирования неисправностей. 4) Разработан кубитно-векторный метод синтеза тестов, использующий производные, вычисляемые по векторным покрытиям логики. 5) Выполнена верификация моделей и методов на тестовых примерах в программной реализации структур и алгоритмов.

Выводы. Научная новизна заключается в новой парадигме технологии синтеза дедуктивной RTL-логики на основе метрического уравнения тестирования, формирующего хог-отношения между тестом, функцией и неисправностями. Вводится векторная форма описания структур, которая дает возможность применить известные технологии синтеза и анализа тестов логических схем для эффективного решения задач тестирования графовых структур и автоматных моделей цифровых устройств. Практическая значимость отражается в примерах аналитического синтеза дедуктивной логики для функциональных элементов векторного уровня и уровня регистровых передач. Предлагается дедуктивный процессор для моделирования неисправностей, который ориентирован на имплементацию в качестве средства BIST, используемое в режиме онлайн тестирования, моделирования и диагностирования неисправностей для цифровых систем на кристаллах. Предлагается кубитно-векторная форма описания цифровых систем, которая превосходит существующие способы задания вычислительных устройств по метрике: технологичность, компактность, быстрдействие и качество. Разработано программное приложение, которое реализует основные сервисы тестирования, моделирования и диагностирования, используемые в учебном процессе для изучения преимуществ кубитно-векторных структур данных и алгоритмов. Приводятся оценки вычислительной сложности процессов синтеза тестов и дедуктивных формул для логики и их использования при моделировании неисправностей.

КЛЮЧЕВЫЕ СЛОВА: RTL логика, синтез, техническая диагностика, моделирование цифровых схем, X-функция, метрика сходства-различия, дедуктивная функция, метрика уравнения тестирования.

VECTOR-DRIVEN LOGIC AND STRUCTURE FOR TESTING AND DEDUCTIVE FAULT SIMULATION

Hahanova A. – PhD, Associated Professor of Design Automation Department, Kharkov National University of Radio Electronics, Ukraine.

Hahanov V. – D. Sc., Professor of Design Automation Department, Kharkov National University of Radio Electronics, Ukraine.
Chumachenko S. – D. Sc., Professor of Design Automation Department, Kharkov National University of Radio Electronics, Ukraine.
Litvinova E. – D. Sc., Professor of Design Automation Department, Kharkov National University of Radio Electronics, Ukraine.
Rakhlis D. – PhD, Associated Professor of Design Automation Department, Kharkov National University of Radio Electronics, Ukraine.

ABSTRACT

Context. It is known that data structures are decisive for the creation of efficient parallel algorithms and high-performance computing devices. Therefore, the development of mathematically perfect and technologically simple data structures takes about 80 percent of the design time, when about 20 percent of time and material resources are spent on algorithms and their hardware-software coding. This leads to search for such primitives of data structures that will significantly simplify the parallel high-performance algorithms which are working on them. Models and methods for testing and simulation of digital systems are proposed, which contain certain advantages of quantum computing in terms of implementation of vector qubit data structures in technology of classical computational processes.

Objective. The goal of the work is development of an innovative technology for qubit-vector synthesis and deductive analysis of tests for their verification based on vector data structures that greatly simplify algorithms that can be embedded as BIST components in digital systems on chips.

Method. The deductive faults simulation is used to obtain analytical expressions focused on transporting fault lists through a functional or logical element based on the xor-operation, which serves as a measure of similarity-difference between a test, a function and faults which is specified in the same way in one of the formats – a table, graph, equation. A binary vector is proposed as the most technologically advanced primitive of data structures for setting logical functionality for the purpose of parallel synthesis and analysis of digital systems. The parallelism of solving combinatorial problems is a physical property of quantum computing, which in classical computing, for parallel simulation and faults diagnostics, is provided by unitary-coded data structures due to excess memory.

Results. 1) A method of analytical synthesis of deductive logic for functional elements on the gate level and register transfer level has been developed. 2) A deductive processor for faults simulation based on transporting input lists or faults vectors to external outputs of digital circuits was proposed. 3) The qubit-vector form of logic setting and methods of qubit synthesis of deductive equations for faults simulation were described. 4) A qubit-vector method for the tests' synthesis which is using derivatives calculated by vector coverage of logic has been developed. 5) Models and methods verification is performed on test examples in the software implementation of structures and algorithms.

Conclusions. The scientific novelty lies in the new paradigm of the technology for the synthesis of deductive RTL logic based on metric test equation, which forms the. A vector form for structures description is introduced, which makes it possible to apply well-known technologies for the synthesis and analysis of logical circuits tests to effectively solve the problems of graph structures testing and state machine models of digital devices. The practical significance is reflected in the examples of analytical synthesis of deductive logic for functional elements on gate level and register transfer level. A deductive processor for faults simulation which is focused on implementation as a BIST tool, which is used in online testing, simulation and fault diagnosis for digital systems on chips is proposed. A qubit-vector form of the digital systems description is proposed, which surpasses the existing methods of computing devices development in terms of the metric: manufacturability, compactness, speed and quality. A software application has been developed that implements the main testing, simulation and diagnostics services which are used in the educational process to study the advantages of qubit-vector data structures and algorithms. The computational complexity of synthesis processes and deductive formulas for logic and their usage in fault simulation are given.

KEYWORDS: RTL logic, test synthesis, technical diagnostics, deductive fault simulation, X-function, similarity-difference metric, deductive function, metric test equation.

REFERENCES

1. Abramovici M., Breuer M. A., Friedman A. D. Digital System Testing and Testable Design. New York, Computer Society Press, 1990, 652 p.
2. Chang H. Y., Chappell S. G., Elmendorf C. H. et al. Comparison of Parallel and Deductive Fault Simulation Methods, *IEEE Transactions on Computers*. November 1974, Vol. C-23, № 11, pp. 1132–1138. DOI: 10.1109/T-C.1974.223820.
3. Takahashi N., Ishiura N., Yajima S. Fault simulation for multiple faults by Boolean function manipulation, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, April 1994, Vol. 13, № 4, pp. 531–535. DOI: 10.1109/43.275363.
4. Menon P. R., Chappell S. G. Deductive Fault Simulation with Functional Blocks, *IEEE Transactions on Computers*, Aug. 1978, Vol. C-27, № 8, pp. 689–695. DOI: 10.1109/TC.1978.1675175.
5. Fujiwara H. Logic Testing and Design for Testability, *Cambridge: MIT Press*, 1985, pp. 84–108.
6. Liu T., Yu T., Wang S. et al. An Efficient Degraded Deductive Fault Simulator for Small-Delay Defects, *IEEE Access*, 2020, Vol. 8, pp. 204855–204862. DOI: 10.1109/ACCESS.2020.3037292.
7. Pomeranz I., Reddy S. M. Forward-looking fault simulation for improved static compaction, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Oct. 2001, Vol. 20, № 10, pp. 1262–1266. DOI: 10.1109/43.952743.
8. Kinoshita Y., Hosokawa T., Fujiwara H. A Test Generation Method Based on k-Cycle Testing for Finite State Machines, *IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS), Rhodes, Greece, 1–3 July 2019: proceedings*. Rhodes, IEEE, 2019, pp. 232–235. DOI: 10.1109/IOLTS.2019.8854426.
9. Harutyunyan G., Tshagharyan G., Vardanian V. et al. Fault modeling and test algorithm creation strategy for FinFET-based memories, *IEEE 32nd VLSI Test Symposium (VTS), Napa, CA, 13–17 April 2014: proceedings*. Napa, IEEE, 2014, pp. 1–6. DOI: 10.1109/VTS.2014.6818747.

10. Zolfy M., Mirkhani S., Navabi Z. Adaptation of an event-driven simulation environment to sequentially propagated concurrent fault simulation, *Design, Automation and Test in Europe. Conference and Exhibition, Munich, Germany, 13–16 March 2001: proceedings*. Munich, IEEE, 2001, pp. 1530–1591. DOI: 10.1109/DATE.2001.915173.
11. Zainalabedin N. Digital System Test and Testable Design using HDL models and architectures. New York, Springer, 2011, 391 p.
12. Pomeranz I., Reddy S. M. Unspecified Transition Faults: A Transition Fault Model for At-Speed Fault Simulation and Test Generation, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Jan. 2008. Vol. 27, № 1, pp. 137–146. DOI: 10.1109/TCAD.2007.907000.
13. Ubar R., Devadze S., J. Raik et al. Fast Fault Simulation for Extended Class of Faults in Scan Path Circuits, *Fifth IEEE International Symposium on Electronic Design, Test & Applications, Ho Chi Minh City, Vietnam, 13–15 Jan. 2010: proceedings*. Vietnam, IEEE, 2010, pp. 14–19. DOI: 10.1109/DELTA.2010.32.
14. Reinsalu U., Raik J., Ubar R. et al. Fast RTL Fault Simulation Using Decision Diagrams and Bitwise Set Operations, *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Vancouver, BC, 3–5 Oct. 2011: proceedings*. Vancouver, IEEE, 2011, pp. 164–170. DOI: 10.1109/DFT.2011.42.
15. Dobai R., Gramatova E. Deductive Fault Simulation for Asynchronous Sequential Circuits, *12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools, Patras, 27–29 Aug. 2009: proceedings*. Patras, IEEE, 2009, pp. 459–464. DOI: 10.1109/DSD.2009.129.
16. Temma S., Sugii M., Matsuno H. The Document Similarity Index based on the Jaccard Distance for Mail Filtering, *34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), JeJu, Korea (South), 23–26 June 2019: proceedings*. JeJu, IEEE, 2019, pp. 1–4. DOI: 10.1109/ITC-CSCC.2019.8793419.
17. Jaccard P. Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines, *Bulletin de la Societe Vaudoise des Sciences. Naturelles*, 1901, Vol. 37, № 140. – P. 241–272. DOI: 10.5169/seals-266440.
18. Hahanov V. Cyber Physical Computing for IoT-driven Services. New York, Springer, 2018, 279 p.
19. Hahanov I., Chumachenko S., Iemelianov I. et al. Deductive qubit fault simulation, *14th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Lviv, 21–25 Feb. 2017: proceedings*. Lviv, IEEE, 2017, pp. 256–259. DOI: 10.1109/CADSM.2017.7916129.
20. Hahanov V., Iemelianov I., Chumachenko S. et al. Quantum sequencer for the minimal test synthesis of black-box functionality, *IEEE East-West Design & Test Symposium (EWDTS), Novi Sad, 29 Sept.–2 Oct. 2017: proceedings*. Novi Sad, IEEE, 2017, pp. 1–6. DOI: 10.1109/EWDTS.2017.8110148.
21. Hahanov V., Yemelyanov I., Obrizan V. et al. “Quantum” diagnosis and simulation of SoC, *XI International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, 2–6 Sept. 2015: proceedings*. Lviv, IEEE, 2015, pp. 58–60.
22. Hahanov V., Liubarskyi M., Gharibi W. et al. Test Synthesis for Logical X-functions, *IEEE East-West Design & Test Symposium (EWDTS), Kazan, 14–17 Sept. 2018: proceedings*. Kazan, IEEE, 2018, pp. 1–9. DOI: 10.1109/EWDTS.2018.8524863.
23. Gharibi W., Devadze D., Hahanov V. et al. Qubit Test Synthesis Processor for SoC Logic, *IEEE East-West Design & Test Symposium (EWDTS), Batumi, Georgia, 13–16 Sept. 2019: proceedings*. Batumi, IEEE, 2019, pp. 1–5. DOI: 10.1109/EWDTS.2019.8884476.
24. Hahanov V. I., Hyduke S. M., Gharibi W. et al. Quantum Models and Method for Analysis and Testing Computing Systems, *11th International Conference on Information Technology: New Generations, Las Vegas, NV, 7–9 April 2014: proceedings*. Las Vegas, IEEE, 2014, pp. 430–434. DOI: 10.1109/ITNG.2014.125.
25. Karavay M., Hahanov V., Litvinova E. et al. Qubit Fault Detection in SoC Logic, *IEEE East-West Design & Test Symposium (EWDTS), Batumi, Georgia, 13–16 Sept. 2019: proceedings*. Batumi, IEEE, 2019, pp. 1–7. DOI: 10.1109/EWDTS.2019.8884475.
26. Hahanov V., Hacimahmud A. V., Litvinova E. et al. Quantum Deductive Simulation for Logic Functions, *IEEE East-West Design & Test Symposium (EWDTS), Kazan, 14–17 Sept. 2018: proceedings*. Kazan, IEEE, 2018, pp. 1–7. DOI: 10.1109/EWDTS.2018.8524619.
27. Hahanov V., Karavay M., Sergienko V. et al. Similarity-Difference Analysis and Matrix Fault Diagnosis of SoC-components, *IEEE East-West Design & Test Symposium (EWDTS), Varna, Bulgaria, 4–7 Sept. 2020: proceedings*. Varna, IEEE, 2020, pp. 1–5. DOI: 10.1109/EWDTS50664.2020.9224740.
28. Hahanov V., Gharibi W., Litvinova E. et al. Qubit-driven Fault Simulation, *IEEE Latin American Test Symposium (LATS), Santiago, Chile, 11–13 March 2019: proceedings*. Santiago, IEEE, 2019, pp. 1–7. DOI: 10.1109/LATW.2019.8704583.
29. Hahanov V., Amer T. B., Litvinova E. et al. Qubit test synthesis of the functionality, *14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Lviv, 21–25 Feb. 2017: proceedings*. Lviv, IEEE, 2017, pp. 251–255. DOI: 10.1109/CADSM.2017.7916128.
30. Hahanov I., Chumachenko S., Iemelianov I. et al. Deductive qubit fault simulation, *14th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Lviv, 21–25 Feb. 2017: proceedings*. Lviv, IEEE, 2017, pp. 256–259. DOI: 10.1109/CADSM.2017.7916129.