

ВЕКТОРНО-ЛОГІЧНЕ МОДЕЛЮВАННЯ НЕСПРАВНОСТЕЙ

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

Чумаченко С. В. – д-р техн. наук, професор, завідувач кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

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

Хаханова Г. В. – д-р техн. наук, професор кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Хаханов І. В. – канд. техн. наук, асистент кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Рожнова Т. Г. – канд. техн. наук, доцент кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Обрізан В. І. – канд. техн. наук, докторант кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

АНОТАЦІЯ

Актуальність. Технологічні тренди Design&Test комп'ютерингу для IT-індустрії та академічної науки завтрашнього дня визначаються такими напрямками: in-memory комп'ютеринг, імерсійний комп'ютеринг, AI-комп'ютеринг, орієнтованими на енергозбереження та скорочення часу обчислень при наданні сервісів. Пропонується механізм моделювання несправностей, як адрес, на розумних структурах даних, які виключають алгоритм моделювання вхідних тестових наборів для отримання тестової карти для логічної функціональності. Запропонований механізм орієнтований на сервісне обслуговування SoC IP-cores під керуванням стандарту IEEE 1500, що може бути сприйнято позитивно інженерами на EDA-ринку.

Мета. Мета дослідження – економічні за часом та енерговитратами механізми моделювання несправностей, як адрес, за рахунок використання read-write транзакцій in-memory комп'ютерингу для побудови карти тестування будь-якої функціональності на розумних структурах даних.

Метод. Розумні структури даних представлені логічним вектором та його похідними у вигляді таблиць істинності та матриць. Карта тестування є матрицею, координати якої визначені комбінаціями всіх логічних несправностей, які перевіряються на двійкових наборах вичерпного тесту. Побудова карти тестування орієнтована на архітектуру in-memory комп'ютерингу на основі read-write транзакцій, що робить механізм моделювання економічним до часу моделювання та енерговитрат завдяки відсутності центрального процесора. Логічний вектор як єдиний компонент вхідних даних не вимагає синтезу в технологічно дозволену структуру елементів. Синтез розумних структур даних на основі чотирьох матричних операцій створює карту тестування несправностей, як адрес, для будь-якої логіки.

Результати. Вектори дедуктивної матриці ефективно використовуються для моделювання несправностей, як адрес, у цифрових структурах будь-якої конфігурації, включаючи розгалуження, що сходяться, і зворотні зв'язки. Отримана карта тестування використовується для знаходження мінімального тесту перевірки несправностей вхідних змінних. Запропонований механізм моделювання несправностей технологічно легко вписується в архітектуру in-memory комп'ютерингу та використовує тільки read-write транзакції. Векторно-логічний механізм можна також використовувати для тестування графових структур, які описуються таблицею істинності або логічним вектором. Адреси таблиці істинності, що використовуються для моделювання несправностей, ефективно застосовуються для безпроцесорної обробки великих даних в архітектурі in-memory комп'ютерингу.

Висновки. Наукова новизна – пропонується механізм векторно-логічного in-memory комп'ютерингу побудови карти тестування, що характеризується побудовою розумних структур даних, які обнулюють алгоритм моделювання несправностей. За простотою та передбачуваністю розмірів структур даних та відсутністю алгоритму моделювання тестових наборів запропонований механізм не має аналогів у design & test індустрії. Практична значимість визначається застосуванням механізму для тестування логічних функціональностей будь-якої складності на вирішення завдань верифікації. Перспективи дослідження – збільшення об'єкта діагностування до схеми, тобто побудова карти тестування схемної логічної структури.

КЛЮЧОВІ СЛОВА: Intelligent Computing, In-memory computing, логічний вектор, логічна матриця, карта тестування, структури даних, векторно-логічне моделювання, несправність, таблиця істинності, адреси.

АБРЕВІАТУРИ

DAC61 – 61st Design Automation Conference;

EDA – Electronic Design Automation;

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

DRAM – Dynamic Random Access Memory (динамічна пам'ять із довільним доступом);

FLASH – Flash Memory (флеш-пам'ять);

RRAM – Resistive Random-Access Memory (резистивна пам'ять із довільним доступом);

PCM – Phase-Change Memory (пам'ять із зміною фазового стану);

MRAM – Magnetoresistive Random-Access Memory (магніторезистивна оперативна пам'ять);

FinFET-fin field-effect transistor (ребристий польовий транзистор);

AI – Artificial Intelligence (штучний інтелект);

DFX – Design for Excellence (проекування для досконалості);

IFS – Intel Foundry Services (послуги Intel з ливарного виробництва);

SLM – Silicon Lifecycle Management (управління життєвим циклом кремнію);

ATPG – Automatic Test Pattern Generation (автоматична генерація тестових шаблонів);

SoC – System-on-Chip (система на кристалі).

НОМЕНКЛАТУРА

L – логічна матриця;

Y – двійковий вектор логічної функціональності;

F – карта тестування логічної функціональності (F -матриця);

A – адреси таблиці істинності (комбінацій логічних несправностей);

H – матриці перекодування;

T – тестові набори;

D – дедуктивна матриця;

A^l – 1-біти;

n – кількість змінних;

w – довжина регістру (слова).

ВСТУП

У 2024 році конференція DAC61 [1] збрала 337 наукових праць з 29 напрямків. Цього року кількість поданих дослідницьких робіт з усіх напрямків та з усього світу зросла на 34%, а також було зафіксовано рекордну кількість поданих 1545 заявок – це після небувалого рекорду за кількістю заявок, встановленого минулого року. Було відзначено, що окрім традиційних тем проєкування EDA, IP та вбудованих систем, з'явилися три додаткові теми – штучний інтелект, автономні системи у пам'яті та безпека. Кількість доповідей щодо застосування штучного інтелекту для проєкування чіпів, архітектури апаратного та програмного забезпечення за останні кілька років буквально зросла у рази. Тут представлені наукові інновації, що важливо, закінчуються інженерними методиками та додатками.

Для ефективного проєкування, тестування, валідації інтегральних схем та компонентів важливо мати просту та розширену мову опису робочого процесу. Сьогодні Python став стандартною мовою програмування для машинного навчання, наукових обчислень та інженерії. Існує також проблема пам'яті. Незважаючи на всі дискусії про закон Мура, одне можна сказати напевно: пам'ять масштабується не так сильно, як логіка. Програми AI, що настільки популярні в наші дні, вимагають все більшого обсягу швидкодіючої та дешевої пам'яті для організації in-memory комп'ютерингу та зберігання даних.

До дослідницької програми Research Track було додано програму Engineering Track, призначену для

галузових практиків та технічних менеджерів, яка фокусується на чотирьох ключових областях: інтерфейсне проєкування, внутрішнє проєкування, інтелектуальна власність, вбудовані системи та програмне забезпечення. Тут інженери галузі діляться останніми інноваціями та ключовими досягненнями. Цього року кількість заявок збільшилась на 32 %, при цьому особлива увага приділялася штучному інтелекту, дизайну та інтелектуальній власності. Декілька десятків доповідей та workshop були присвячені створенню комп'ютерних архітектур у пам'яті. Особливо ця тема звучала при реалізації Artificial Intelligence моделей, економних за енерговитратами та часом. Тут використовуються такі типи пам'яті: SRAM, DRAM, FLASH, RRAM, PCM, MRAM, або FinFET-Nanosheet as their memory technology. Інтеграція AI з in-memory computing надає людству нові можливості економіки великих даних (рис. 1).

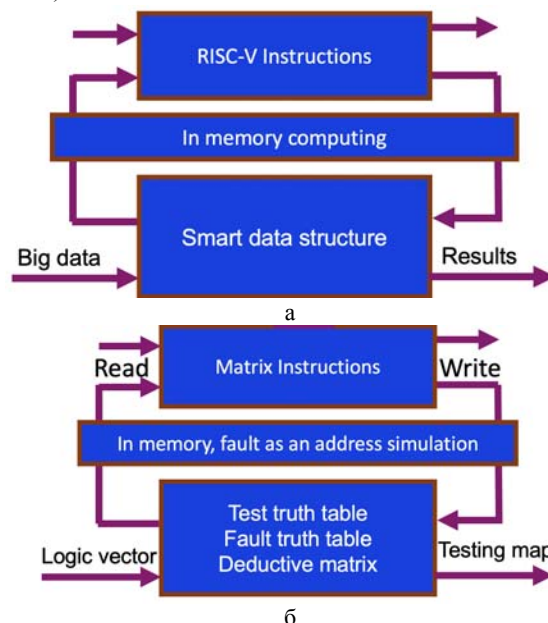


Рисунок 1 – Структура in-memory комп'ютерингу обробки даних та моделювання несправностей: а – комп'ютеринг у пам'яті; б – моделювання несправності у пам'яті як адрес

На рис. 1 позначено: In-memory computing – комп'ютеринг у пам'яті; Smart Data Structure – розумні структури даних; RISC-V Instructions – інструкції обчислювача з набором спрощених/редукованих команд (Reduced instruction set computer); розумні структури даних; Matrix Instructions – матричні інструкції; In-memory, fault as address simulation – моделювання несправності пам'яті як адрес; Test truth table – таблиця істинності тесту; Fault truth table – таблиця істинності несправності; Deductive matrix – дедуктивна матриця; Big Data – великі дані; Results – результати; Read – читання; Write – запис; Logic vector – логічний вектор; Testing map – карта тестування.

Об'єкт дослідження – in-memory intelligente комп'ютинг, який знижує енергетичні та часові витрати під час обробки великих даних.

Предмет дослідження – in-memory аналіз елементів чи цифрових схем будь-якої розмірності за допомогою read-write транзакцій на логічних векторах.

Будь-які інженерні рішення, орієнтовані на збереження енергії та часу проектування, завжди будуть потрібні на ринку EDA. Виходячи з цього, можна сформулювати мету дослідження – економічні за часом та енерговитратами механізми моделювання несправностей, як адрес, за рахунок використання read-write транзакцій in-memory комп'ютингу для побудови карти тестування будь-якої функціональності на розумних структурах даних.

1 ПОСТАНОВКА ЗАДАЧІ

Розглядається задача моделювання несправностей як адрес на розумних структурах даних, які виключають алгоритм моделювання вхідних тестових наборів для отримання карти тестування логічної функціональності.

Нехай задано двійковий вектор логічної функціональності Y . Слід побудувати карту тестування F логічної функціональності Y . Практичне завдання полягає у створенні векторно-логічного механізму моделювання несправностей.

Задачі, що підлягають розв'язку: 1) визначення розумних структур даних, що функціонально залежать від логічного вектора; 2) визначення мінімального числа операцій для суперпозиції розумних та явних структур даних, що дозволяють без алгоритму моделювання вхідних тестових наборів визначати карту тестування логічної функціональності; 3) верифікація механізмів моделювання несправностей як адрес, на прикладах логічних функціональностей.

2 ОГЛЯД ЛІТЕРАТУРИ

Конференція DAC61 [1] є щорічною помітною подією у світі, на яких обговорюються топ-технології глобального комп'ютингу сьогодення. Це дуже важливо як для коригування програм академічної науки, так і використання AI-пілотів для освоєння курсів студентами [2]. DAC61 23–27 червня 2024 року проходив у Сан-Франциско під прапором «The Chip to Systems», запропонованим компанією Intel. Як зменшити шлях між чіпом та системою для IT-індустрії. Як виготовляти чіпи, орієнтовані на системи. Як швидко заливати системи у виготовлені чіпи та чіплети. Як тестувати та верифікувати системи, що налічують трильйон транзисторів [3]. Крім традиційних тем проектування EDA, IP та вбудованих систем, з'явилися три додаткові важливі теми: штучний інтелект, автономні іммерсивні системи та безпека. Кількість доповідей із штучного інтелекту

зросла в рази. Вони були присвячені вирішенню питань проектування та верифікації чіпів, апаратного забезпечення та архітектури. DAC61 запропонував трендові чотири чудові доповіді, присвячені штучному інтелекту, створенню обчислювальних систем у пам'яті та іммерсивним обчисленням, а також кілька цікавих виступів SKYtalks та Techtalks, що охоплюють широкий спектр тем, пов'язаних з мікросхемами, системами та додатками. Декілька яскравих прикладів сказаного.

Andrew B. Kahng, Professor UC San Diego: «Останні роки принесли потік рішень щодо штучного інтелекту в EDA. Потенційні переваги AI-EDA включає покращену якість проектування, аналізу та моделювання з меншими часовими, матеріальними та енергетичними витратами [4–11]. Це не залишилося непоміченим як в академічних колах, так і промисловості».

Dr. Gary Patton, Intel, представив концепцію створення «ливарної системи» – чіплети, наповнені програмним забезпеченням. Робиться це шляхом проектування високопродуктивних обчислювачів за епоху штучного інтелекту. Пропозиція зробити екосистему «EDA-IP» має життєво важливе значення для створення 3D-систем на основі штучного інтелекту.

CEO Intel Пат Гелсінгер заявив, що Intel Foundry Services (IFS) відкриває «еру створення систем». Замість того, щоб просто постачати клієнтам пластини, що є традиційною моделлю ливарного виробництва, Intel вже сьогодні пропонує пластини кремнію, корпуси, програмне забезпечення та чіплети. «IFS відкриє еру створення систем, – сказав він, – відзначаючи зміну парадигми, оскільки фокус переміщається від систем на кристалі до «систем-в-корпусі» (SoP) або чіплетам».

Yervant Zorian, Synopsys. Кремній за своєю суттю ненадійний, а пристрої та вузли на його основі найбільш сприйнятливі до відмов. DFX – Design for Excellence на просунутих вузлах потребує нових стратегій. У доповіді розглядалися механізми відмов, методи керування, що забезпечують надійний удосконалений вузол кремнію. Ключові теми включають безпеку, захищеність, надійність та SLM – Silicon Lifecycle Management. Методології та структури управління життєвим циклом кремнію, які необхідні для забезпечення надійності обчислювальних систем.

Sarita V. Adve Professor, University of Illinois. Іммерсивні чи просторові обчислення – це загальна концепція низки технологій, які оцифровують діяльність машин, механізмів, людей та об'єктів, а також середовище, в якому вони відбуваються (рис. 2). Створення центру іммерсивних обчислень IMMERSE, який поєднує іммерсивні технології, програми та людський досвід. Показано проект ILLIXR, який базується на наскрізній системі з відкритим вихідним кодом для демократизації

досліджень іммерсивних систем та їх впровадження до ІТ-ринку. Імерсивний комп'ютинг вирішує проблеми метричного управління оцифрованими віртуальними, фізичними та соціальними процесами [7–9].

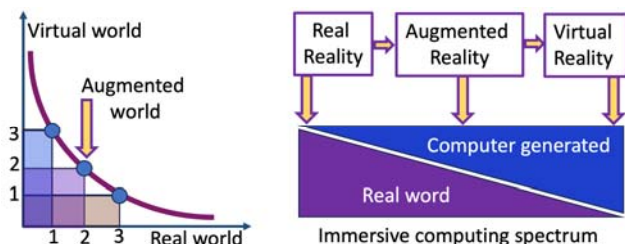


Рисунок 2 – Два подання імерсивного комп'ютингу: Virtual world – віртуальний світ; Augmented world – доповнений світ; Real world – реальний світ; Real Reality – реальна (існуюча) реальність; Augmented Reality – доповнена реальність; Virtual Reality – віртуальна реальність; Computer generated – згенеровано комп'ютером; Immersive computing spectrum – спектр імерсивного комп'ютингу

Alan Lee, Chief Technology Officer, Analog Devices, Штучний інтелект змінює навколишній світ, але основна увага приділяється великим моделям, що працюють на величезних обчислювальних потужностях. Існує гостра потреба в штучному інтелекті в периферійних додатках на основі архітектури in-memory комп'ютингу для зменшення затримок та енергоспоживання [8–11]. Задоволення цієї потреби вимагає нових підходів, що дозволяють задовольнити обмеження майбутніх промислових, автомобільних та споживчих платформ на інтелектуальній периферії. Вартість експлуатації будь-якого комп'ютера – це насамперед вартість електроенергії, спожитої цим пристроєм.

Модельовання несправностей дуже важливе [12–16] для АТПГ, верифікації, діагностики, класифікації несправностей. Метрикою ефективності (табл. 1) промислових методів моделювання несправностей є швидкість, пам'ять, складність алгоритмів моделювання функціональних блоків, затримок, елементів, послідовних схем, багатозначне

модельовання несправностей, обробка невідомих сигналів X та високоімпедансних Z .

Враховується складність створення моделей, алгоритмів моделювання елементів та структур. Зручність форми подання моделі справної поведінки та несправностей. Промисловий лідер – спільне моделювання несправностей – це кероване моделювання з використанням good/bad подій. В одному кадрі моделюється підмножина несправних схем, які відрізняються від справної поведінки пристрою. Тут виникають проблеми управління пам'яттю. Практично всі промислові системи моделювання несправностей [14–18] мають непередбачуваний розмір списку несправностей та непередбачуваний розмір структур даних. Усі шість основних методів моделювання використовують процесор із високим рівнем енергоспоживання. Механізм векторного моделювання [19–24] несправностей виграє перед промисловими аналогами по всіх пунктах, крім одного – не враховуються затримки елементів.

3 МАТЕРІАЛИ І МЕТОДИ

Суть дослідження, доведеного рівня інженера, полягає у паралельному моделюванні несправностей логіки на вичерпному тесті без алгоритму моделювання вхідних наборів. Інакше кажучи, пропонується адресне моделювання несправностей логіки на інтелектуальних структурах даних без алгоритму моделювання (рис. 3).

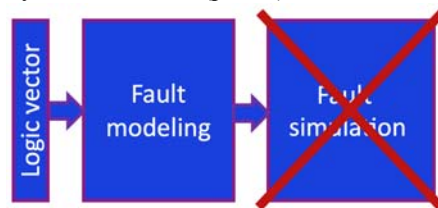


Рисунок 3 – Моделювання несправностей без алгоритму моделювання

Побудова моделі розумних структур даних використовує чотири послідовні процедури синтезу наступних матриць, що формують рішення (рис. 4).

Таблиця 1 – Порівняння промислових технологій моделювання несправностей

Fault simulation technique	Complexity	Memory	Data structure	Level	Delay	Speed	Fault model	Multy-valued
Serial fault simulation (1963)	$n \times n^3$	Predictable	Add fault model, fault list	Gate, system	No problem	Slowest	Any	Easy
Parallel fault simulation (1965)	$\frac{1}{w} \times n^3$	Predictable	Register Memory	Gate	Not capable	Middle	Logic	Dificult
Deductive fault simulation (1972)	n^2	Unpredictable	Deductive formulas	Gate	Not capable	Middle	Any	Dificult
Concurrent fault simulation (1974)	$\frac{1}{3} \times n^2$	Unpredictable	Add fault model, fault list	Gate, RTL	Capable	Faster	Logic	Easy
PPSFP – Parallel pattern single fault propagation (1985)	$\frac{1}{w} \times n^3$	Unpredictable	Add fault model, fault list	Gate	Capable	Middle	Logic	Easy
Differential fault simulation (1989)	$\frac{1}{2} \times n^2$	Unpredictable	Add fault model, fault list	Gate, RTL	Not capable	Middle	Any	Dificult
Vector fault simulation (2023)	$\frac{1}{2} \times \frac{1}{3} \times n^2$	Predictable	No, As true-value simulation	Gate, RTL, System	Not capable	Faster	Logic	Capable

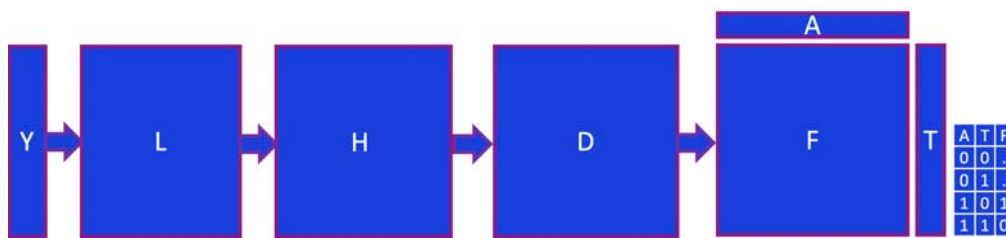


Рисунок 4 – Синтез картки тестування без алгоритму моделювання

1) Синтез логічної L -матриці шляхом взяття декартового хог-квадрата на бітах логічного вектора від n -змінних за формулою: $L=Y \text{ хог } Y = Y^2_{\text{XOR}}$.

2) Побудова матриці перекодування H шляхом взяття декартового хог-квадрата на адресах таблиці A істинності від n змінних за формулою: $H=A \text{ хог } A = A^2_{\text{XOR}}$. Адреси виконують роль тестових наборів T та виступають комбінаціями логічних несправностей A . Отримана матриця є константою для всіх логічних функцій від n змінних.

3) Створення дедуктивної матриці D шляхом переадресації координат логічної L -матриці на H -матриці перекодування за такою формулою: $D=L_H$.

4) Отримання карти тестування або матриці несправностей, що перевіряються на вичерпному тесті, шляхом виконання координатних операцій на 1-бітах A^1 з A векторів таблиці істинності в 1-координатах дедуктивної D -матриці за формулою: $F=A^1(-T)$ або $F=A^1 \text{ хог } T$. Знаки F несправностей, що перевіряються, вхідних змінних в координатах карти тестування F визначаються інверсією бітів тестових вхідних наборів T . Таблиця істинності для формування знаків перевірених несправностей на координатах F -матриці наведена в правій частині (див. рис. 4).

4 ЕКСПЕРИМЕНТИ

Далі наводиться фрагмент (рис. 5) структури формування знаків несправностей, що перевіряються в алфавіті $\{0,1, \llcorner, \gg\}$ на координатах карти тестування, відповідних 1-координатам дедуктивної матриці.

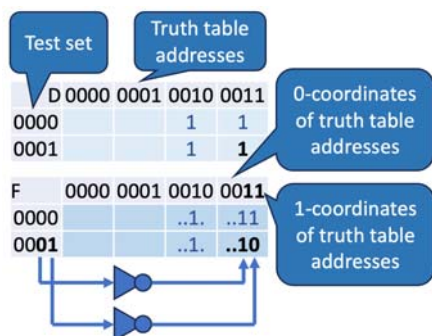


Рисунок 5 – Формування координат карти тестування

Несправності, що перевіряються, формуються тільки по одиничних координатах адрес таблиці істинності. Ці 1-координати визначаються інверсними значеннями бітів двійкових тестових наборів. 0-

координати адрес таблиці істинності в карті тестування довідзначаються точками, які означають відсутність несправностей, що перевіряються на цих вхідних змінних. Синтез усієї карти тестування функцій від чотирьох змінних буде наведено нижче (рис. 9). Найпростіша реалізація векторно-логічного моделювання несправностей як адрес починається з функціональностей, що мають одну змінну. На основі логічних векторів (00, 01, 10, 11) будуються дедуктивні матриці та карти тестування несправностей (рис. 6). Оскільки логічні вектори 01,10 та 00,11 є взаємно-інверсними, то вони генерують еквівалентні логічні та дедуктивні матриці та карти тестування.

Logic vector 00				
L	H	D	F	
Y 0 0			0 1	0 1
0	0 1	0		0
0	1 0	1		1
Logic vector 01				
L	H	D	F	
Y 0 1			0 1	0 1
0 1	0 1	0 1	0	1
1 1	1 0	1 1	1	0
Logic vector 10				
L	H	D	F	
Y 1 0			0 1	0 1
1 1	0 1	0 1	0	1
0 1	1 0	1 1	1	0
Logic vector 11				
L	H	D	F	
Y 1 1			0 1	0 1
1	0 1	0		0
1	1 0	1		1

Рисунок 6 – Синтез карт тестування для логіки з однією змінною

Несправності першої та останньої логічної функції не можуть бути перевірені, тому що логічні вектори 00 та 11 не мають змін сигналів. Щоб з'явилися несправності, які можуть бути перевірені, повинні бути логічні вектори, в яких є хоча б один нуль і хоча б одна одиниця.

Наступним пунктом експериментів на механізмі моделювання є побудова карт тестування для кількох логічних функцій двох змінних (рис. 7).

Ці матриці можна використовувати для аналізу несправностей логічних схем з урахуванням стандартних технологічно дозволених елементів. Крім того, їх побудова є навчальною вибіркою для студентів, які бажають освоїти векторно-логічне моделювання несправностей як адрес таблиці істинності логічної функціональності.

6 ОБГОВОРЕННЯ

Перевагою даного векторно-логічного механізму моделювання несправностей як адрес таблиці істинності є новий науковий результат, доведений до рівня інженерного розуміння – як побудувати карту тестування або швидко написати код для її автоматичного синтезу. Основний науковий інноваційний результат – це використання адрес таблиці істинності як тестових вхідних наборів та комбінації логічних несправностей. Судячи з публікацій, цього ніхто досі не робив. Перспектива використання адрес таблиці істинності – це обробка великих даних у пам'яті, де вони зберігаються, за алгоритмом лінійної обчислювальної складності. Обмеження векторно-логічного механізму тестування несправностей пов'язані з візуалізацією матриць і карт тестування від великого (>10) числа логічних змінних.

Ринкова привабливість запропонованого дослідження визначається наступною метрикою, цікавою для EDA-ринку:

1. Економію обробки великих даних як адрес in-memo комп'ютерингу, вільним від процесора [5–7], що дає можливість економити електроенергію (36%) і час обробки великих даних (22%).

2. Суттєвим зниженням обчислювальної складності запропонованих алгоритмів аналізу логічних схем за рахунок експоненційної надмірності розумних структур даних.

3. Інваріантністю запропонованих векторно-логічних структур даних для моделювання несправностей як адрес вентиляного, реєстрового та системного рівня подання проектів.

4. Передбачуваністю розмірів пам'яті для зберігання розумних структур даних та обчислювальної складності алгоритмів для моделювання несправностей логіки та цифрових схем.

5. Мінімальною кількістю даних про проект для синтезу тестів на основі моделювання логічних несправностей, як адрес.

6. Використання чотирьох векторно-матричних паралельних операцій для синтезу карти тестування логічних функціональностей.

7. Відкриттям тестової константи – універсальної матриці логічних несправностей як відстаней між тестовими наборами та адресами таблиці істинності логічної функції, що у кілька разів зменшує складність алгоритмів моделювання несправностей.

8. Векторно-логічна модель структури дозволяє тестувати несправності переходів будь-якого графа розробленими механізмами синтезу карти тестування. За допомогою кодованого унітарного даного на універсумі примітивів можна побудувати вектор логічної моделі будь-якого соціального процесу, потім його протестувати розробленими механізмами.

ВИСНОВКИ

Узагальнюючи отримані результати з верифікації векторно-логічного механізму моделювання несправностей, можна зробити такі наукові та практичні висновки: 1) імплементація механізму моделювання в коді мови Python займає 400 рядків; 2) логічного вектора достатньо для побудови карти тестування функціональності, він є суто простою та ефективною інженерною методикою для верифікації SoC IP-cores; 3) звичайно, що питання часу обробки великих за розміром логічних векторів тут не стоїть. Час їхньої обробки – мілісекунди. Проблема полягає лише у візуалізації великих за розміром карт тестування логічних функціональностей, що мають $n > 10$ змінних; 4) вектори дедуктивної матриці ефективно використовуються для моделювання несправностей, як адрес, у цифрових структурах будь-якої конфігурації, включаючи розгалуження, що сходяться, і зворотні зв'язки; 5) отримана карта тестування використовується для знаходження мінімального тесту перевірки несправностей вхідних змінних; 6) даний механізм моделювання несправностей технологічно легко вписується в архітектуру in-memo комп'ютерингу і використовує лише read-write транзакції; 7) векторно-логічний механізм можна використовувати для тестування графових структур, які описуються таблицею істинності або логічним вектором; 8) адреси таблиці істинності, що використовуються для моделювання несправностей, ефективно застосовуються для безпроцесорної обробки великих даних в архітектурі in-memo комп'ютерингу.

Наукова новизна – пропонується механізм векторно-логічного in-memo комп'ютерингу побудови карти тестування, що характеризується створенням розумних структур даних, які обнулюють алгоритм моделювання несправностей.

Практична значимість визначається застосуванням механізму для тестування логічних функціональностей будь-якої складності на вирішення завдань верифікації.

Перспективи дослідження – збільшення об'єкта діагностування до схеми, тобто побудова карти тестування схемної логічної структури.

ЛІТЕРАТУРА

1. Design Automation Conference «The chips to the systems» [Electronic resource]. Access mode: https://www.dac.com/Portals/0/DAC%2061/Program/DAC_1352254-24_Digital-Onsite-Program-7.pdf?ver=hfG38JtEimK1HdWaAtD_g%3d%3d
2. News: Professors Rethink. How They Teach Coding Students embrace AI copilots; teachers shift to problem-solving [Text] / [R. D. Caballar, A. Jones, D. Genkina and C. Q. Choi] // IEEE Spectrum. – Vol. 61, № 7. – P. 5–12. DOI: 10.1109/MSPEC.2024.10589683.
3. Liu M. The Path to a 1-Trillion-Transistor GPU: AI's Boom Demands New Chip Technology [Text] / M. Liu and H.-S.

- P. Wong // IEEE Spectrum. – 2024. – Vol. 61, № 7. – P. 22–27. DOI: 10.1109/MSPEC.2024.10589682.
4. MLiM: High-Performance Magnetic Logic in-Memory Scheme with Unipolar Switching SOT-MRAM [Text] / [B. Wu, H. Zhu, K. Chen et al.] // IEEE Transactions on Circuits and Systems I: Regular Papers. – 2023. – Vol. 70, №6. – P. 2412–2424. DOI: 10.1109/TCSI.2023.3254607.
 5. Testing for Electromigration in Sub-5nm FinFET Memories [Text] / [M. Mayahinia et al.] // IEEE Design & Test. – 2024. – June. – P. 1. DOI 10.1109/MDAT.2024.3411527.
 6. Resistance-Sum Architecture for Voltage-Controlled SOT-MRAM based Computing-in-Memory with Hybrid References [Text] / [C. Xiao et al.] // 2023 IEEE International Magnetic Conference-Short Papers (INTERMAG Short Papers), Sendai, Japan. – 15–19 May 2023: proceedings. – Sendai: IEEE. – P. 1–2. DOI: 10.1109/INTERMAGShortPapers58606.2023.10228265.
 7. AI Accelerator Embedded Computational Storage for Large-Scale DNN Models [Text] / [B. Ahn, J. Jang, H. Na et al.] // 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Incheon, Korea, Republic, 13–15 June 2022: proceedings. – Incheon: IEEE, 2022. – P. 483–486. DOI: 10.1109/AICAS4282.2022.9869991.
 8. OPC: A Distributed Computing and Memory Computing-Based Efficient Solution of Big Data [Text] / [Z. Yang, C. Zhang, M. Hu, and F. Lin] // 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015: proceedings. – Chengdu: IEEE, 2015. – P. 50–53. DOI: 10.1109/SmartCity.2015.46.
 9. Reliable ReRAM-based Logic Operations for Computing in Memory [Text] / [M. Moreau et al.] // 2018 IFIP/IEEE International Conference on Very Large-Scale Integration (VLSI-SoC), Verona, Italy, 08–10 October 2018: proceedings. – Verona: IEEE, 2018. – P. 192–195. DOI: 10.1109/VLSI-SoC.2018.8644780.
 10. Kang W. Spintronic Memories: From Memory to Computing-in-Memory [Text] / W. Kang, H. Zhang, and W. Zhao // 2019 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), Qingdao, China, 17–19 July 2019: proceedings. – Qingdao: IEEE, 2019. P. 1–2. DOI: 10.1109/NANOARCH47378.2019.181298.
 11. Memory Sizing of a Scalable SRAM In-Memory Computing Tile Based Architecture [Text] / [R. Gauchi et al.] // 2019 IFIP/IEEE 27th International Conference on Very Large-Scale Integration (VLSI-SoC), Cuzco, Peru, 06–09 October 2019: proceedings. – Cuzco: IEEE, 2019. – P. 166–171. DOI: 10.1109/VLSI-SoC.2019.8920373
 12. Ulrich E. G. Exclusive Simulation of Activity in Digital Networks [Text] / E. G. Ulrich // Communications of the ACM (CACM). – 1969. – Vol. 13, №2. – P. 102–110. DOI: 10.1145/362848.362870
 13. Armstrong D. B. A Deductive Method for Simulating Faults in Logic Circuits [Text] / D. B. Armstrong // IEEE Trans. on Computers. – 1972. – Vol. C-21. – P. 464–471. DOI: 10.1109/T-C.1972.223542
 14. Ulrich E. G. The Concurrent Simulation of Nearly Identical Digital Networks [Text] / E. G. Ulrich, and T. Baker // 10th Design Automation Workshop, June 25 – 27, 1973: proceedings. – USA: IEEE Press, 1973. – P. 145–150. <https://dl.acm.org/doi/epdf/10.5555/800124.804009>
 15. Donald M. Schuler. Random test generation using concurrent logic simulation [Text] / Donald M. Schuler, Ernst G. Ulrich, Thomas E. Baker, and Susan P. Bryant // 12th Design Automation Conference (DAC '75). – USA: IEEE Press, 1975. – P. 261–267. <https://dl.acm.org/doi/abs/10.5555/800261.809076>
 16. Abramovici M. Digital Systems Testing and Testable Design / M. Abramovici, M.A. Breuer, A.D. Friedman. New York: IEEE Press, 1990. 657 p. DOI:10.1109/9780470544389
 17. Chul Young Lee. PROBE: a PPSFP simulator for resistive bridging faults / Chul Young Lee and D.M.H. Walker // 18th IEEE VLSI Test Symposium, Montreal, Quebec, Canada, 30 April 2000 – 04 May 2000: proceedings. – P. 105–110. DOI: 10.1109/VTEST.2000.843833.
 18. Riahi, Navabi. A VPI-based combinational IP core module-based mixed level serial fault simulation and test generation methodology [Text] / Riahi, Navabi, and Lombardi // 2003 Test Symposium, Xi'an, China, 16–19 November 2003: proceedings. – IEEE Press^ 2003. – P. 274–277 DOI: 10.1109/ATS.2003.1250822.
 19. Vector-logic computing for faults-as-address deductive simulation [Text] / [W. Gharibi, V. Hahanov, S. Chumachenko et al.] // IAES International Journal of Robotics and Automation (IJRA). – 2023. – Vol. 12, № 3. – P. 274–288. DOI: 10.11591/ijra.v12i3.p.274-288.
 20. In-Memory Intelligent Computing [Text] / [V.I. Hahanov, V. H. Abdullayev, S. V. Chumachenko, E. I. Lytvynova, I. V. Hahanova] // Radio Electronics, Computer Science, Control. – 2024. – №1. – P. 161–174. DOI: 10.15588/1607-3274-2024-1-15.
 21. Vector-Logical Fault Simulation [Text] / [V. Hahanov, S. Chumachenko, E. Litvinova et al.] // Radio Electronics, Computer Science, Control. – 2023. – № 2. – P. 37–51. DOI: 10.15588/1607-3274-2023-2-5.
 22. Vector-Driven Logic and Structure for Testing and Deductive Fault Simulation [Text] / [A. Hahanova, V. Hahanov, S. Chumachenko et al.] // Radio Electronics, Comput. Science, Control. – 2021. – №3. – P. 69–85. DOI: 10.15588/1607-3274-2021-3-7.
 23. Vector-Deductive Memory-Based Transactions for Fault-as-Address Simulation [Text] / [W. Gharibi, A. Hahanova, V. Hahanov et al.] // Electronic modeling. – 2023. – №45(1). – P. 3–26. DOI: 10.15407/emodel.45.01.003.
 24. Vector-Logic Synthesis of Deductive Matrices for Fault Simulation [Text] / [W. Gharibi, A. Hahanova, V. Hahanov et al.] // Electronic modeling. – 2023. – №45(2). – P. 16–33. DOI: 10.15407/emodel.45.02.016.

Стаття надійшла до редакції 17.09.2024.

Після доробки 07.11.2024.

VECTOR-LOGIC FAULT SIMULATION

Hahanov V. I. – Dr. Sc., Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Chumachenko S. V. – Dr. Sc., Professor, Head of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Lytvynova E. I. – Dr. Sc., Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Khakhanova H. V. – Dr. Sc., Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Hahanov I. V. – PhD, Assistant of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Rozhnova T. G. – PhD, Associate Professor of the the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Obrizan V. I. – PhD, Post-Doctoral Student of the the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

ABSTRACT

Context. The technological trends of Design&Test computing for the IT industry and academic science are determined by the following directions: in-memory computing, immersive computing, AI computing, focused on energy saving and reduction of computing time when providing services. A mechanism for simulating faults as addresses on smart data structures is proposed, which eliminates the algorithm for simulating input test sets to obtain a test map for logic functionality. The proposed mechanism is focused on the service of SoC IP-cores under the control of the IEEE 1500 standard, which can be perceived positively by engineers in the EDA market.

Objective. The purpose of the research is time- and energy-saving mechanisms for simulating malfunctions, such as addresses, by using read-write transactions of in-memory computing to build a test map of any functionality on smart data structures.

Method. Smart data structures are represented by a logical vector and its derivatives in the form of truth tables and matrices. The test map is a matrix whose coordinates are determined by the combinations of all logical faults that are tested on the binary sets of the comprehensive test. The construction of the test map is focused on the architecture of in-memory computing based on read-write transactions, which makes the simulation mechanism economical in terms of simulation time and energy consumption due to the absence of a central processor. A logical vector as a single component of input data does not require synthesis into a technologically permitted structure of elements. Synthesis of smart data structures based on four matrix operations creates a fault test map like addresses for any logic.

Results. Deductive matrix vectors are effectively used to model faults as addresses in digital structures of any configuration, including convergent branches and feedback loops. The resulting test map is used to find the minimum fault-checking test of the input variables. The proposed fault simulation mechanism technologically easily fits into the architecture of in-memory computing and uses only read-write transactions. The vector logic engine can also be used to test graph structures that are described by a truth table or a logical vector. The truth table addresses used for fault simulation are effectively used for processorless processing of large data in the in-memory computing architecture.

Conclusions. Scientific novelty – a vector-logic in-memory computing mechanism for building a test map is proposed, characterized by the construction of intelligent data structures that reset the fault modeling algorithm. The proposed mechanism has no analogues in the design & test industry in terms of simplicity and predictability of data structure sizes and the absence of a test set modeling algorithm. The practical significance is determined by the application of the mechanism for testing logical functionalities of any complexity to solve verification tasks. Prospects of the research – increasing the object of diagnosis to the scheme, i.e. building a test map of the scheme logical structure.

KEYWORDS: Intelligent Computing, In-Memory Computing, Logic Vector, Logic Matrix, Test Map, Data Structures, Vector-Logic Modeling, Fault, Truth Table, Addresses.

REFERENCE

1. Design Automation Conference «The chips to the systems» [Electronic resource]. Access mode: https://www.dac.com/Portals/0/DAC%2061/Program/DAC_1352254-24_Digital-Onsite-Program-7.pdf?ver=hfG38JtJEimK1HdWaAtD_g%3d%3d
2. Caballar R. D., Jones A., Genkina D. and Choi C. Q. News: Professors Rethink. How They Teach Coding Students embrace AI copilots; teachers shift to problem-solving [Text], *IEEE Spectrum*, Vol. 61, № 7, pp. 5–12. DOI: 10.1109/MSPEC.2024.10589683.
3. Liu M. and Wong H.-S. P. The Path to a 1-Trillion-Transistor GPU: AI's Boom Demands New Chip Technology [Text], *IEEE Spectrum*, 2024, Vol. 61, № 7, pp. 22–27. DOI: 10.1109/MSPEC.2024.10589682.
4. Wu B., Zhu H., Chen K., Yan C., and Liu W. MLiM: High-Performance Magnetic Logic in-Memory Scheme with Unipolar Switching SOT-MRAM [Text], *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023, Vol. 70, №6, pp. 2412–2424. DOI: 10.1109/TCSI.2023.3254607.
5. Mayahinia, M. et al. Testing for Electromigration in Sub-5nm FinFET Memories [Text], *IEEE Design & Test*, 2024, June, P. 1. DOI 10.1109/MDAT.2024.3411527.
6. Xiao C. et al. Resistance-Sum Architecture for Voltage-Controlled SOT-MRAM based Computing-in-Memory with Hybrid References [Text], *2023 IEEE International Magnetic Conference-Short Papers (INTERMAG Short Papers), Sendai, Japan, 15–19 May 2023: proceedings*. Sendai, IEEE, pp. 1–2. DOI: 10.1109/INTERMAGShortPapers58606.2023.10228265.

© Хаханов В. І., Чумаченко С. В., Литвинова Є. І., Хаханова Г. В., Хаханов І. В., Рожнова Т. Г.,
Обрізан В. І., 2024
DOI 10.15588/1607-3274-2024-4-18



7. Ahn B., Jang J., Na H., Seo M., Son H. and Song Y. H. AI Accelerator Embedded Computational Storage for Large-Scale DNN Models [Text], *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Incheon, Korea, Republic, 13–15 June 2022: proceedings*. Incheon, IEEE, 2022, pp. 483–486. DOI: 10.1109/AICASS4282.2022.9869991.
8. Yang Z., Zhang C., Hu M., and Lin F. OPC: A Distributed Computing and Memory Computing-Based Efficient Solution of Big Data [Text], *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015: proceedings*. Chengdu, IEEE, 2015, pp. 50–53. DOI: 10.1109/SmartCity.2015.46.
9. Moreau M. et al. Reliable ReRAM-based Logic Operations for Computing in Memory [Text], *2018 IFIP/IEEE International Conference on Very Large-Scale Integration (VLSI-SoC), Verona, Italy, 08–10 October 2018: proceedings*. Verona, IEEE, 2018, pp. 192–195. DOI: 10.1109/VLSI-SoC.2018.8644780.
10. Kang W. Zhang H., and Zhao W. Spintronic Memories: From Memory to Computing-in-Memory [Text], *2019 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), Qingdao, China, 17–19 July 2019: proceedings*. Qingdao, IEEE, 2019, pp. 1–2. DOI: 10.1109/NANOARCH47378.2019.181298.
11. Gauchi R. et al. Memory Sizing of a Scalable SRAM In-Memory Computing Tile Based Architecture [Text], *2019 IFIP/IEEE 27th International Conference on Very Large-Scale Integration (VLSI-SoC), Cuzco, Peru, 06–09 October 2019: proceedings*. Cuzco, IEEE, 2019, pp. 166–171. DOI: 10.1109/VLSI-SoC.2019.8920373
12. Ulrich E. G. Exclusive Simulation of Activity in Digital Networks [Text], *Communications of the ACM (CACM)*, 1969, Vol. 13, №2, pp. 102–110. DOI: 10.1145/362848.362870
13. Armstrong D. B. A Deductive Method for Simulating Faults in Logic Circuits [Text], *IEEE Trans. on Computers*, 1972, Vol. C-21. – P. 464–471. DOI: 10.1109/T-C.1972.223542
14. Ulrich E. G. and Baker T. The Concurrent Simulation of Nearly Identical Digital Networks [Text], *10th Design Automation Workshop, June 25–27, 1973: proceedings*. USA, IEEE Press, 1973, pp. 145–150. <https://dl.acm.org/doi/epdf/10.5555/800124.804009>
15. Donald M. Schuler, Ernst G. Ulrich, Thomas E. Baker, and Susan P. Bryant Random test generation using concurrent logic simulation [Text], *12th Design Automation Conference (DAC '75)*. USA, IEEE Press, 1975, pp. 261–267. <https://dl.acm.org/doi/abs/10.5555/800261.809076>
16. Abramovici M., Breuer M. A., Friedman A. D. Digital Systems Testing and Testable Design. New York, IEEE Press, 1990. 657 p. DOI:10.1109/9780470544389
17. Chul Young Lee and Walker D. M. H. PROBE: a PPSFP simulator for resistive bridging faults, *18th IEEE VLSI Test Symposium, Montreal, Quebec, Canada, 30 April 2000 – 04 May 2000: proceedings*, pp. 105–110. DOI: 10.1109/VTEST.2000.843833.
18. Riahi, Navabi and Lombardi A VPI-based combinational IP core module-based mixed level serial fault simulation and test generation methodology [Text], *2003 Test Symposium, Xi'an, China, 16–19 November 2003: proceedings*, IEEE Press, 2003, pp. 274–277 DOI: 10.1109/ATS.2003.1250822.
19. Gharibi W., Hahanov V., Chumachenko S., Litvinova E., Hahanov I., Hahanova I. Vector-logic computing for fault-as-address deductive simulation [Text], *IAES International Journal of Robotics and Automation (IJRA)*, 2023, Vol. 12, № 3, pp. 274–288. DOI: 10.11591/ijra.v12i3.P.274–288.
20. Hahanov V. I., Abdullayev V. H., Chumachenko S. V., Lytvynova E. I., Hahanova I. V. In-Memory Intelligent Computing [Text], *Radio Electronics, Computer Science, Control*, 2024, №1, pp. 161–174. DOI: 10.15588/1607-3274-2024-1-15.
21. Hahanov V., Chumachenko S., Litvinova E., Hahanova I., Khakhanova A., Shkil A., Rakhlis D., Hahanov I., Shevchenko O. Vector-Logical Fault Simulation [Text], *Radio Electronics, Computer Science, Control*, 2023, №2, pp. 37–51. DOI: 10.15588/1607-3274-2023-2-5.
22. Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Rakhlis D. Vector-Driven Logic and Structure for Testing and Deductive Fault Simulation [Text], *Radio Electronics, Comput. Science, Control*, 2021, №3, pp. 69–85. DOI: 10.15588/1607-3274-2021-3-7.
23. Gharibi W., Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Hahanov I. Vector-Deductive Memory-Based Transactions for Fault-as-Address Simulation [Text], *Electronic modeling*, 2023, №45(1), pp. 3–26. DOI: 10.15407/emodel.45.01.003.
24. Gharibi, W. Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Hahanov I. Vector-Logic Synthesis of Deductive Matrices for Fault Simulation [Text], *Electronic modeling*, 2023, №45(2), pp. 16–33. DOI: 10.15407/emodel.45.02.016.