

DOI: <https://doi.org/10.15276/ict.01.2024.26>

УДК 681.326

Моделювання та симуляція в пам'яті

Хаханов Володимир Іванович¹⁾Д-р техн. наук., професор каф. Автоматизації проектування обчислювальної техніки
ORCID: <https://orcid.org/0000-0001-5312-5841>; hahanov@icloud.com**Чумаченко Світлана Вікторівна¹⁾**Д-р техн. наук., професор, завідувач каф. Автоматизації проектування обчислювальної техніки
ORCID: <https://orcid.org/0000-0001-8913-1194>; svetlana.chumachenko@nure.ua**Литвинова Євгенія Іванівна¹⁾**Д-р техн. наук., професор каф. Автоматизації проектування обчислювальної техніки
ORCID: <https://orcid.org/0000-0002-9797-5271>; eugenia.litvinova@nure.ua**Обрізан Володимир Ігорович¹⁾**Канд. техн. наук, докторант каф. Автоматизації проектування обчислювальної техніки
ORCID: <https://orcid.org/0000-0002-1835-4056>; vladimir.obrizan@nure.ua¹⁾ Харківський національний університет радіоелектроніки, пр. Науки, 14. Харків, 61166, Україна

АНОТАЦІЯ

Пропонується механізм моделювання несправностей, як адрес, на розумних структурах даних, які виключають алгоритм моделювання вхідних тестових наборів для отримання тестової карти логічної функціональності. Розумні структури даних представлені логічним вектором та його похідними як таблиці істинності та матриці. Карта тестування подана матрицею, координати якої визначені комбінаціями всіх логічних несправностей, які перевіряються на двійкових наборах вичерпного тесту. Побудова карти тестування орієнтована на архітектуру in-memoгу комп'ютерингу на основі read-write транзакцій, що робить механізм моделювання економічним по відношенню до часу моделювання та енерговитрат завдяки відсутності центрального процесора. Логічний вектор як єдиний компонент вхідних даних не вимагає синтезу в технологічно дозволу структуру елементів. Синтез розумних структур даних на основі чотирьох матричних операцій створює карту тестування несправностей, як адрес, для будь-якої логіки. Запропонований механізм орієнтований на сервісне обслуговування SoC IP-cores під керуванням стандарту IEEE 1500. За простотою та передбачуваністю розмірів структур даних та відсутністю алгоритму моделювання тестових наборів запропонований механізм не має аналогів у design and test індустрії.

Ключові слова: Intelligent комп'ютеринг; In-memoгу комп'ютеринг; логічний вектор; логічна матриця; карта тестування; структури даних; векторно-логічне моделювання; несправність; таблиця істинності; адреси

Метою дослідження є створення економічних за часом та енерговитратами механізмів моделювання несправностей, як адрес, за рахунок використання транзакцій запису-зчитування in-memoгу комп'ютерингу для побудови карти тестування довільної функціональності на розумних структурах даних.

Конференція Design Automation Conference у 2024 році [1] інтегрувала 337 наукових праць з 29 тематичних напрямів. Окрім традиційної тематики проектування EDA, IP та вбудованих систем, виникли три додаткові теми – штучний інтелект, автономні системи у пам'яті та безпека. Число доповідей, що присвячено застосуванню штучного інтелекту для проектування чіпів, архітектурі апаратного та програмного забезпечення, зросла у рази за останні кілька років. Представлені наукові інновації, що закінчуються інженерними методиками та додатками. Визначено корисні технологічні тренди DAC61 розвитку комп'ютерингу для IT-індустрії та академічної науки найближчого майбутнього. Виділяються такі напрямки: in-memoгу комп'ютеринг, імерсивний комп'ютеринг, AI-комп'ютеринг, які орієнтовані на енергозбереження та скорочення часу обчислень при наданні сервісів. Потенційні переваги AI-EDA містять покращену якість проектування, аналізу та моделювання з меншими часовими, матеріальними та енергетичними витратами [2-9]. Механізм векторного моделювання [10-15] несправностей виграють перед промисловими аналогами за всіма пунктами, крім одного – не враховуються затримки елементів.

Для ефективного проектування, тестування, валідації інтегральних схем та компонентів важливо мати просту та розширену мову опису робочого процесу. Такою стандартною мовою програмування для машинного навчання, наукових обчислень та інженерії сьогодні став Python.

This is an open-access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/deed.uk>)

Існує також проблема пам'яті. Незважаючи на всі дискусії про закон Мура, можна сказати напевно: пам'ять масштабується не так сильно, як логіка. Програми штучного інтелекту, що настільки популярні в наші дні, вимагають все більшого обсягу швидкодіючої та дешевої пам'яті для організації in-меморі комп'ютерингу та зберігання даних. Декілька десятків доповідей 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.

Інтеграція штучного інтелекту з in-меморі комп'ютерингом надає людству нові можливості економіки великих даних (Рис. 1):

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.

Сутність дослідження полягає у паралельному моделюванні несправностей логіки на вичерпному тесті без алгоритму моделювання вхідних наборів. Пропонується адресне моделювання несправностей логіки на інтелектуальних структурах даних саме без алгоритму моделювання.

Побудова моделі розумних структур даних використовує чотири послідовні процедури синтезу наступних матриць, що формують рішення (Рис. 2).

1) Синтез логічної L-матриці шляхом взяття декартового \oplus -квадрату на бітах логічного вектора від n-змінних за формулою:

$$L = Y \oplus Y = Y_{\oplus}^2. \quad (1)$$

2) Побудова матриці перекодування H шляхом взяття декартового \oplus -квадрата на адресах таблиці A істинності від n змінних за формулою:

$$H = A \oplus A = A_{\oplus}^2. \quad (2)$$

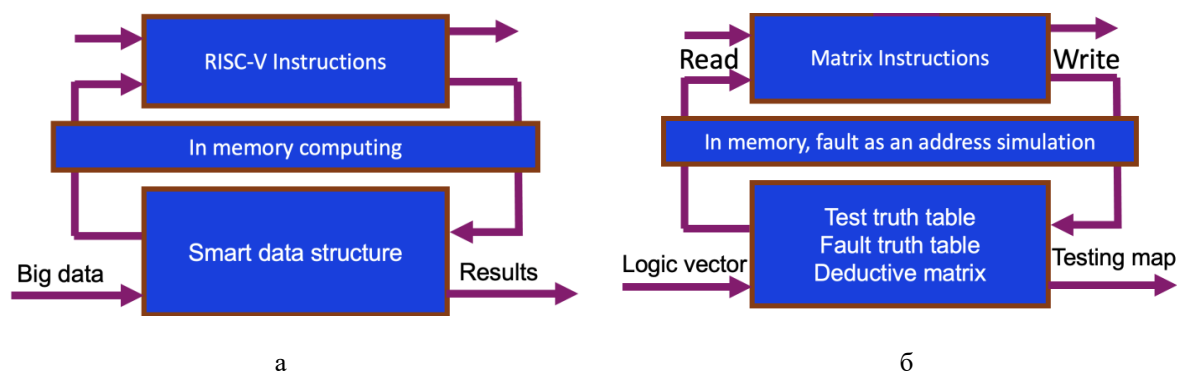


Рис. 1. Структура in-меморі комп'ютерингу обробки даних та моделювання несправностей:

а – комп'ютеринг у пам'яті; б – моделювання несправності у пам'яті як адрес

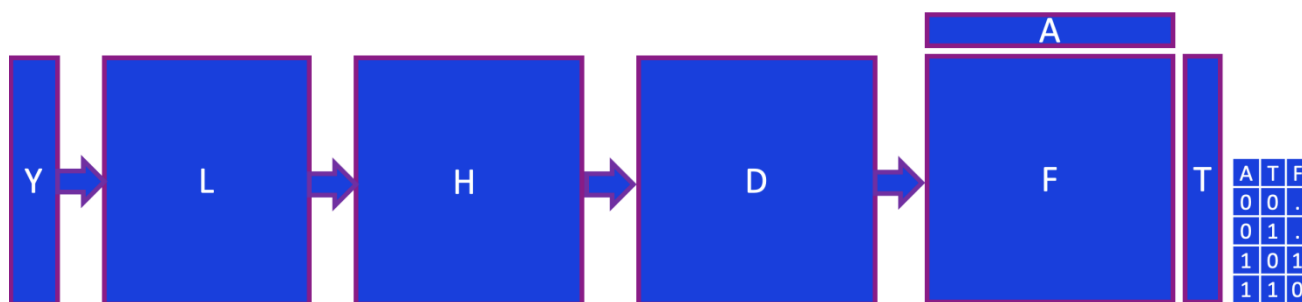


Рис. 2 . Синтез картки тестування без алгоритму моделювання

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 $A^1 \in A$ векторів таблиці істинності в 1-координатах дедуктивної D -матриці за формулою:

$$F=A^1\bar{T} \text{ or } F=A^1\oplus T. \quad (3)$$

Знаки F несправностей, що перевіряються, на входних змінних в координатах карти тестування F визначаються інверсією бітів тестових входних наборів T . Таблиця істинності для формування знаків перевірених несправностей на координатах F -матриці наведена в правій частині Рис. 2.

Несправності, що перевіряються, формуються тільки за одиничними координатами адрес таблиці істинності. Ці 1-координати визначаються інверсними значеннями бітів двійкових тестових наборів. 0-координати адрес таблиці істинності в карті тестування доводяться точками, які означають відсутність несправностей, що перевіряються на цих входних змінних. Найпростіша реалізація векторно-логічного моделювання несправностей як адрес починається з функціональностей, що мають одну змінну. Оскільки логічні вектори 01.10 та 00,11 є взаємно-інверсними, то вони генерують еквівалентні логічні та дедуктивні матриці та карти тестування.

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. Далі виконується побудова карти тестування для логічних функцій від трьох змінних. Заключним пунктом експериментів на механізмі моделювання є побудова карти тестування для логічної функції 0011111111111100 від чотирьох змінних (Рис. 3).

Таким чином, запропоновано механізм векторно-логічного in-методу комп'ютерного 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.

СПИСОК ЛІТЕРАТУРИ

1. “Design Automation Conference “The chips to the systems””. – Available from: https://www.dac.com/Portals/0/DAC%2061/Program/DAC_1352254-24_Digital-Onsite-Program-7.pdf?ver=hfG38JtJEimK1HdWaAtD_g%3d%3d.
- 2 Wu B., Zhu H., Chen K., Yan C., and Liu W. “MLiM: High-Performance Magnetic Logic in-Memory Scheme with Unipolar Switching SOT-MRAM”. *IEEE Transactions on Circuits and Systems I: Regular Papers*. 2023; 70 (6): 2412–2424. DOI: <https://doi.org/10.1109/TCSI.2023.3254607>.
3. Mayahinia M., et al. “Testing for Electromigration in Sub-5nm FinFET Memories”. *IEEE Design & Test*. 2024. DOI: <https://doi.org/10.1109/MDAT.2024.3411527>.
4. Xiao C. et al. “Resistance-Sum Architecture for Voltage-Controlled SOT-MRAM based Computing-in-Memory with Hybrid References”. *IEEE International Magnetic Conference-Short Papers (INTERMAG Short Papers)*. Sendai, Japan. 2023. p. 1–2. DOI: <https://doi.org/10.1109/INTERMAGShortPapers58606.2023.10228265>.
5. Ahn B., Jang J., Na H., Seo M., Son H. and Song Y. H. “AI Accelerator Embedded Computational Storage for Large-Scale DNN Models”. *IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. Incheon, Korea Republic. 2022. p. 483–486. DOI: <https://doi.org/10.1109/AICAS54282.2022.9869991>.
6. Yang Z., Zhang C., Hu M., and Lin F. “OPC: A Distributed Computing and Memory Computing-Based Efficient Solution of Big Data”. *IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*. Chengdu, China. 2015. p. 50–53. DOI: <https://doi.org/10.1109/SmartCity.2015.46>.
7. Moreau. M. et al. “Reliable ReRAM-based Logic Operations for Computing in Memory”. *IFIP/IEEE International Conference on Very Large-Scale Integration (VLSI-SoC)*. Verona, Italy. 2018. p. 192–195. DOI: <https://doi.org/10.1109/VLSI-SoC.2018.8644780>.
8. Kang W., Zhang H., and Zhao W. “Spintronic Memories: From Memory to Computing-in-Memory”. *IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*., Qingdao, China. 2019. p. 1–2. DOI: <https://doi.org/10.1109/NANOARCH47378.2019.181298>.
9. Gauchi R. et al. “Memory Sizing of a Scalable SRAM In-Memory Computing Tile Based Architecture”. *IFIP/IEEE 27th International Conference on Very Large-Scale Integration (VLSI-SoC)*. Cuzco, Peru. 2019. p. 166–171. DOI: <https://doi.org/10.1109/VLSI-SoC.2019.8920373>.
10. Gharibi W., Hahanov V., Chumachenko S., Litvinova E., Hahanov V. I., Hahanova I. “Vector-logic computing for faults-as-address deductive simulation”. *IAES International Journal of Robotics and Automation (IJRA)*. 2023; 12 (3): 274–288. DOI: <https://doi.org/10.11591/ijra.v12i3.p.274-288>.
11. Hahanov V. I., Abdullayev V. H., Chumachenko S. V., Lytvynova E. I., Hahanova I. V. “In-Memory Intelligent Computing”. *Radio Electronics, Computer Science, Control*. 2024. 1: 161174. DOI: <https://doi.org/10.15588/1607-3274-2024-1-15>.
12. Hahanov V., Chumachenko S., Litvinova E., Hahanova I., Khakhanova A., Shkil A., Rakhlis D., Hahanov I., Shevchenko O. “Vector-Logical Fault Simulation”. *Radio Electronics, Computer Science, Control*. 2023; 2: 37–51. DOI: <https://doi.org/10.15588/1607-3274-2023-2-5>.
13. Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Rakhlis D. “Vector-Driven Logic and Structure for Testing and Deductive Fault Simulation”. *Radio Electronics, Comput. Science, Control*. 2021; 3: 69–85. DOI: <https://doi.org/10.15588/1607-3274-2021-3-7>.
14. Gharibi W., Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Hahanov I. “Vector-Deductive Memory-Based Transactions for Fault-as-Address Simulation”. *Electronic modeling*. 2023; 45 (1): 3–26. DOI: <https://doi.org/10.15407/emodel.45.01.003>.

15. Gharibi W., Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Hahanov I. “Vector–Logic Synthesis of Deductive Matrices for Fault Simulation”. *Electronic modeling*. 2023; 45 (2): 16–33. DOI: <https://doi.org/10.15407/emodel.45.02.016>.

DOI: <https://doi.org/10.15276/ict.01.2024.26>

UDC 681.326

In-Memory modelling and simulations

Vladimir I. Hahanov¹⁾

Dr. Sc., Professor, Design Automation Department

ORCID: <https://orcid.org/0000-0001-5312-5841>; hahanov@icloud.com

Svitlana V. Chumachenko¹⁾

Dr. Sc., Professor, head of Design Automation Department

ORCID: <https://orcid.org/0000-0001-8913-1194>; svetlana.chumachenko@nure.ua

Eugenia I. Litvinova¹⁾

Dr. Sc., Professor, professor, Design Automation Department

ORCID: <https://orcid.org/0000-0002-9797-5271>; eugenia.litvinova@nure.ua

Volodymyr I. Obrizan¹⁾

PhD, Post-Doc Student, Design Automation Department

ORCID: <https://orcid.org/0000-0002-1835-4056>; vladimir.obrizan@nure.ua

¹⁾ Kharkiv National University of Radioelectronics, Nauky Ave, 14. Kharkiv, 61166, Ukraine

ABSTRACT

A mechanism for modeling faults as addresses on smart data structures is proposed, which excludes the algorithm for modeling input test sets to obtain a test map of logical functionality. 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 defined by the combinations of all logical faults that are tested on the binary sets of the exhaustive 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. The proposed mechanism is focused on the service of SoC IP-cores under the control of the IEEE 1500 standard. The proposed mechanism has no analogues in the design and test industry in terms of simplicity and predictability of data structure sizes and the absence of a test set modeling algorithm.

Keywords: Intelligent computing; In-memory computing; logic vector; logic matrix; test map; data structures; vector-logic modeling; fault; truth table; addresses