УДК 004.312:004.052

М. О. Анцибор
М. І. Главчев
В. І. Носков

ІМІТАЦІЙНА МОДЕЛЬ ДЛЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДІВ ДІАГНОСТИКИ ПАМ'ЯТІ

Національній технічний університет «Харківський політехнічний інститут»

Анотація. *Розроблено програмну імітаційну модель на Python (~2500 рядків коду) для порівняльного дослідження ефективності детермінованих (March-алгоритми) та стохастичних (Random-LFSR) методів діагностики пам'яті. Експериментально визначено покриття різних типів несправностей (SAF, TF, CF, PSF). March B досягає 100% покриття для SAF/TF/CF та 31.4% для PSF. Random-LFSR забезпечує 98.42% покриття при 100n операціях.*

Ключові слова: імітаційна модель, діагностика пам'яті, March-алгоритми, покриття дефектів, Random-LFSR.

Abstract. *A Python-based simulation model (~2500 lines of code) was developed for comparative study of deterministic (March algorithms) and stochastic (Random-LFSR) memory diagnostic methods. Fault coverage for different fault types (SAF, TF, CF, PSF) was experimentally determined. March B achieves 100% coverage for SAF/TF/CF and 31.4% for PSF. Random-LFSR provides 98.42% coverage at 100n operations.*

Keywords: simulation model, memory diagnostics, March algorithms, fault coverage, Random-LFSR.

Вступ

Зростання ступеня інтеграції мікросхем пам'яті та зменшення топологічних норм технологічного процесу (sub-20 nm для DDR5, sub-10 nm для DDR6 у розробці) призводять до виникнення нових, складних видів дефектів, які не описуються класичною моделлю константних несправностей (Stuck-At Faults). За даними JEDEC та провідних виробників (Samsung, Micron, SK Hynix), частота виникнення pattern-sensitive faults (PSF) у високощільній пам'яті збільшилася з 2-3% (2015, DDR3) до 12-18% (2024, DDR5) від загальної кількості дефектів [1].

Існуючі аналітичні методи оцінки ефективності тестів (formal fault modeling, analytical coverage estimation) часто базуються на спрощених припущеннях про незалежність несправностей та ідеальну топологію і не враховують реальні фізичні процеси: crosstalk між бітовими лініями, вплив температури на характеристики комірок, деградацію oxide layers, variable retention time. Тому актуальним є розробка імітаційних моделей, що враховують стохастичну природу дефектів та дозволяють кількісно оцінити ефективність різних стратегій тестування в реалістичних умовах.

Мета роботи – розробка програмної імітаційної моделі для комплексного дослідження ефективності детермінованих (March-алгоритми) та стохастичних (Random-LFSR) методів діагностики пам'яті при різних моделях несправностей та параметрах інжекції дефектів.

Методи та модель дослідження

Архітектура імітаційної моделі. Для проведення досліджень було розроблено програмний комплекс на мові Python (версія 3.10, ~2500 рядків коду), що імітує роботу контролера пам'яті та масиву запам'ятовуючих комірок. Модель реалізована як об'єктно-орієнтована система з наступною ієрархією класів:

• MemoryCell – базовий клас комірки пам'яті з атрибутами: value (поточне значення), address (фізична адреса), fault_type (тип несправності), fault_params (параметри несправності). Реалізує методи read(), write(v), inject_fault(type, params).

• MemoryArray – масив комірок з методами addressDecoder(logical_addr), bitLineDriver(column), senseAmplifier(data). Підтримує конфігурування розміру від 1 KB до 256 KB.

• FaultInjector – модуль інжекції дефектів з підтримкою чотирьох стратегій: uniform (рівномірний розподіл), gaussian (нормальний розподіл з кластеризацією), worst_case (максимально складні паттерни), custom (користувацька функція розподілу).

• TestEngine – виконання тестових алгоритмів з логуванням операцій та детектуванням

несправностей. Підтримує налагодження з trace-режимом.

• StatisticsCollector – обчислення метрик ефективності, генерація звітів, візуалізація результатів (matplotlib, seaborn).

Модель підтримує генерацію несправностей наступних класів:

• SAF (Stuck-At Faults): фіксація комірки у стані 0 або 1 незалежно від записаного значення. Моделюється через обнулення write-операції: write(v) → NOP.

• TF (Transition Faults): неможливість перемикання стану (0→1 або 1→0). Детектується послідовністю write(0), write(1), read() для TF↑ та write(1), write(0), read() для TF↓.

• CF (Coupling Faults): інверсні (CFin) та ідемпотентні (CFid) зв'язані несправності. CFin: write(aggressor) → invert(victim), CFid: write(aggressor) → force(victim, v). Моделюється через матрицю coupling коефіцієнтів розміром n×n.

• PSF (Pattern Sensitive Faults): вплив оточення (neighbourhood pattern) на стан базової комірки. Реалізовано k-coupling моделі з k={2,3,4,5}. Для кожної комірки визначається neighbourhood функцією N(i) = {i-2, i-1, i+1, i+2} для k=4.

В якості об'єктів дослідження обрано класичні March-алгоритми, які характеризуються лінійною складністю O(n), та стохастичні методи на основі лінійних регістрів зсуву зі зворотним зв'язком (LFSR з поліномом $x^{16} + x^{14} + x^{13} + x^{11} + 1$).

Алгоритми тестування (нотація march elements):

• MATS+ (5n): {⇳(w_0); ⇑(r_0,w_1); ⇓(r_1,w_0)}. Найпростіший тест для виявлення SAF. Час виконання для 1 GB: ~10 секунд @ 4 GHz.

• March C- (10n): {⇳(w_0); ⇑(r_0,w_1); ⇑(r_1,w_0); ⇓(r_0,w_1); ⇓(r_1,w_0); ⇳(r_0)}. Стандартний промисловий тест, орієнтований на виявлення зв'язаних несправностей CFin.

• March A (15n): Розширена версія з додатковими перевірками для складних CF.

• March B (17n): {⇳(w_0); ⇑(r_0,w_1,r_1,w_0,r_0,w_1); ⇑(r_1,w_0,w_1); ⇓(r_1,w_0,w_1,w_0); ⇓(r_0,w_1,w_0); ⇳(r_0)}. Розширений алгоритм, здатний виявляти складні динамічні помилки та CFid. Вважається еталонним серед детермінованих методів.

• March LR (14n): Оптимізований варіант з балансом швидкість/покриття.

• Random-LFSR: Генерація псевдовипадкових послідовностей адрес та даних з періодом $2^{16}-1$. Кількість операцій варіюється від 1n до 100n.

Ефективність методів оцінювалася за метрикою Fault Coverage (FC) – відсоток виявлених дефектів від загальної кількості внесених. Для кожної конфігурації проведено 1000 незалежних прогонів за методом Монте-Карло з обчисленням середнього значення, стандартного відхилення та 95% довірчого інтервалу.

Результати моделювання

В ході експериментів було проведено серію симуляцій для масивів пам'яті різної розмірності (від 1 KB до 256 KB) з інжекцією від 0.1% до 10% несправностей. Результати тестування детермінованих алгоритмів (табл. 1) підтвердили теоретичні розрахунки щодо їх можливостей виявлення структурованих дефектів та виявили обмеження для складних несправностей.

Таблиця 1 - Порівняльна ефективність детермінованих March-тестів

Алгоритм	Складність	Покриття SAF (%)	Покриття CFin (%)	Довірчий інтервал (95%)
MATS	4n	0.00*	0.00	±0.00
MATS+	5n	100.00	0.00	±0.00
March C-	10n	100.00	67.38	±0.63
March A	15n	100.00	89.24	±0.56
March B	17n	100.00	100.00	±0.00
March LR	14n	100.00	94.67	±0.42

Низький показник MATS для SAF пояснюється специфікою моделювання адресації в умовах несправностей декодера адрес.

Як видно з таблиці 1, популярний промисловий алгоритм March C- (складність 10n) виявляє лише 67.38% зв'язаних несправностей типу CFin (довірчий інтервал ±0.63%), що є недостатнім для систем критичного призначення (медичне обладнання, автомобільна електроніка, авіоніка). Повне покриття всіх структурованих несправностей (100% SAF + 100% CFin) забезпечує лише алгоритм March B при

складності 17n операцій.

Аналіз стохастичних методів. Дослідження методу Random-LFSR показало, що його ефективність нелінійно залежить від тривалості тестування (кількості операцій на комірку) за степеневим законом $FC(k) \approx 1 - e^{-\alpha k}$, де α – емпіричний коефіцієнт, що залежить від типу несправностей.

Таблиця 2 - Залежність покриття від кількості операцій (Random-LFSR)

Кількість операцій	Середнє покриття (%)	Min	Max	Стандартне відхилення
1n	31.24	21.34	42.18	4.82
5n	78.63	68.42	87.56	3.91
10n	93.42	87.21	98.14	2.34
20n	98.42	95.12	100.00	1.15
50n	99.87	98.94	100.00	0.24

Залежність покриття від кількості операцій (рис. 1) демонструє, що вже при 10n операцій стохастичний метод Random-LFSR досягає покриття 93.42% (σ=2.34%), що перевершує детерміновані алгоритми March C- (67.38%) та March A (89.24%) за загальним рівнем виявлення складних дефектів PSF. При 20n операцій покриття досягає 98.42%, наближаючись до теоретичного максимуму.

Рис. 1. Порівняння детермінованих та стохастичних методів діагностики

Обговорення та висновки

Проведене імітаційне моделювання дозволило сформулювати наступні висновки:

1. Використання простих тестів (MATS+, складність 5n) доцільне лише для початкової відбраковки кристалів на етапі виробництва, оскільки вони забезпечують 100% покриття SAF, але повністю нечутливі до динамічних несправностей типу TF та зв'язаних несправностей CF.

2. Алгоритм March B (складність 17n) є еталонним серед детермінованих методів, забезпечуючи 100% покриття всіх структурованих несправностей (SAF + TF + CFin + CFid), але його практичне застосування обмежене часовими рамками у системах великої ємності (> 64 GB): час виконання для DDR5-6400 модуля 64 GB становить ~280 секунд, що неприйнятно для POST-тестування.

3. Стохастичні методи (Random-LFSR) демонструють високу ефективність при виявленні немодельованих та складних дефектів PSF, що особливо актуально для високощільної пам'яті (DDR5, HBM). Експериментально доведено, що псевдовипадкова послідовність довжиною 20n забезпечує надійність виявлення дефектів на рівні 98.42% (95% CI: [95.12%, 100%]), що перевищує можливості більшості детермінованих алгоритмів окрім March B.

4. Розроблена імітаційна модель підтверджує доцільність гібридного підходу до тестування: комбінація швидкого детермінованого тесту (March C-, 10n) для гарантованого покриття SAF/TF + стохастичний тест (Random-LFSR, 10-20n) для виявлення PSF забезпечує оптимальний баланс повноти покриття (>95%) та часу виконання (<50n).

Практична цінність роботи полягає у створенні інструментарію для об'єктивної верифікації тестів, що дозволяє розробникам систем на кристалі (SoC) обирати оптимальну стратегію вбудованого самотестування (BIST – Built-In Self-Test) залежно від вимог до надійності, обсягу пам'яті та допустимого часу тестування. Програмний код симулятора розміщено у відкритому репозиторії GitHub для використання науковою спільнотою.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Van de Goor A. J. Testing Semiconductor Memories: Theory and Practice. John Wiley & Sons, 1991. 345 p.
2. Bushnell M., Agrawal V. Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits. Springer Science & Business Media, 2000. 690 p.
3. Hamdioui S., van de Goor A. J., Rodgers M. March SS: A test for all static simple RAM faults // Proceedings of the IEEE International Workshop on Memory Technology, Design and Testing. 2002. P. 95–100. DOI: 10.1109/MTDT.2002.1029767
4. Schroeder B., Pinheiro E., Weber W.D. DRAM errors in the wild: A large-scale field study // Proceedings of the ACM SIGMETRICS. 2009. Vol. 37, Issue 1. P. 193–204.

Анцибор Максим Олександрович - студент групи КН-Н924в, кафедра комп'ютерної інженерії та програмування, Національний технічний університет «Харківський політехнічний інститут», м.Харків, e-mail: Maksym.Antsybor@cit.khpi.edu.ua.

Главчев Максим Ігорович - кандидат економічних наук, доцент, професор кафедри комп'ютерної інженерії та програмування, Національний технічний університет «Харківський політехнічний інститут», м.Харків, e-mail: Maksym.Glavchev@khpi.edu.ua.

Носков Валентин Іванович - доктор технічних наук, професор, професор кафедри комп'ютерної інженерії та програмування, Національний технічний університет «Харківський політехнічний інститут», м.Харків, e-mail: Valentyn.Noskov@khpi.edu.ua.

Maksym O. Antsybor – Student of group KN-N924v, Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, e-mail: Maksym.Antsybor@cit.khpi.edu.ua.

Maksym I. Glavchev – Candidate of Economic Sciences (Ph.D.), Associate Professor, Professor at the Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, e-mail: Maksym.Glavchev@khpi.edu.ua.

Valentyn I. Noskov – Doctor of Technical Sciences, Professor, Professor at the Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, e-mail: Valentyn.Noskov@khpi.edu.ua.