

DOI: <https://doi.org/10.36910/6775-2524-0560-2023-51-06>

УДК 004.383.3

**Кардашук Володимир Сергійович**<sup>1</sup>, к.т.н., доцент

<https://orcid.org/0000-0002-7940-6753>

**Бортник Катерина Яківна**<sup>2</sup>, к.т.н., доцент

<https://orcid.org/0000-0001-5282-099X>

**Багнюк Наталія Володимирівна**<sup>2</sup>, к.т.н., доцент

<https://orcid.org/0000-0002-7120-5455>

<sup>1</sup> Східноукраїнський національний університет імені Володимира Даля, м. Сєверодонецьк, Україна

<sup>2</sup> Луцький національний технічний університет, м. Луцьк, Україна

## МЕТОДИ ВДОСКОНАЛЕННЯ ТЕСТУВАННЯ ЦИФРОВИХ СИСТЕМ

**Кардашук В.С., Бортник К.Я., Багнюк Н.В. Методи вдосконалення тестування цифрових систем.** Розглянуті дослідження в напрямку технології Design & Test, дедуктивно-паралельне моделювання несправності цифрових систем, проведена класифікація існуючих рішень вдосконалення тестування цифрових систем, запропонована узагальнена модель дедуктивно-паралельного аналізу цифрових систем, представлена внутрішня модель та структура даних для аналізу цифрового пристрою, запропонований дедуктивний аналіз дефектів, розглянутий синтез дедуктивної функції за допомогою таблиці істинності.

**Ключові слова:** SiP, SoC, дефекти, тестування, цифрові системи, синтез, таблиця істинності.

**Kardashuk V., Bortnyk K., Bahniuk N. Methods for improving the testing of digital systems.** The article examines the latest research of Design & Test technology, deductive parallel modeling of the malfunction of digital systems, classifies existing solutions for improving the testing of digital systems, proposes a generalized model of deductive parallel analysis of digital systems, presents an internal model and data structures for the analysis of a digital device, proposes deductive defect analysis, the synthesis of the deductive function with the help of the truth table.

**Keywords:** SiP, SoC, defects, testing, digital systems, synthesis, truth table.

**Вступ.** Домінування цифрових систем на кристалах (SoC – System-on-Chip) та цифрових систем в пакетах (SiP – System-on-Pack) пояснюється тим, що розвиток високих технологій дозволив об'єднати цифрові, аналогові, радіочастотні та навіть компоненти мікроелектромеханічних систем (MEMS – Microelectromechanical systems) в одному пристрої, що зменшує загальний розмір системи та підвищує її продуктивність [1]. На сьогоднішній день створення мініатюрних, спеціалізованих, економічних мобільних та швидкодіючих цифрових виробів є дуже затребуваним у всьому світі. Тестування SoC і SiP необхідне для забезпечення відсутності помилок в продукції через недоліки виробництва. Фактично тестова діяльність відіграє ключову роль у процесі виробництва інтегральних мікросхем завдяки її здатності відрізнити хороші пристрої від несправних перед доставкою кінцевим споживачам, а також для покращення сервісного обслуговування.

В основному тестування складається з двох процесів: генерації тестів і тестового застосування. Виробництво тестів означає процес надання відповідних випробувальних впливів, тоді як тестова програма відноситься до процесу застосування цих тестових впливів до входів схеми та аналізу її реакції. Адаптація існуючих технологій тестування SoC [2], а також розробка нових технологій тестування SiP на сьогоднішній час є пріоритетною проблемою серед вчених та дослідників design-and-test community. Актуальність дослідження полягає в необхідності значного підвищення швидкодії засобів моделювання та генерації тестів для структурно- та функціонально-складних цифрових систем, що імплементуються у кристали програмованої логіки.

**Аналіз останніх досліджень.** Системи тестування фірм Logic Vision, Synopsys, Mentor Graphics, Candence, їх засоби синтезу тестів та моделювання несправностей неприйнятні за часом обробки цифрових систем на кристалах, в яких налічується декілька мільйонів вентилів. В контексті розгляду цифрових виробів важливо диференціювати типи дефектів за ієрархією: одиночні або кратні несправності - для логічних модулів та пам'яті [3], мікродефекти - для компонентів, що входять до складу SoC або SiP [4]. Щодо систем на кристалах, то реалізація вбудованих тестових методів діагностування передбачає наявність інфраструктури сервісного обслуговування. Стратегія тестування SiP відрізняється від SoC, оскільки система SiP інтегрує компоненти різних виробників. Найперспективнішою стратегією для вбудованого тестування функціональних компонентів є використання стандарту IEEE 1500 [5]. З практичної точки зору протягом тривалого часу для вбудованих інструментів тестування стабільно використовується стандарт IEEE 1149.1 (JTAG) [6]

та успішно модифікується в такі стандарти як IEEE 1149.7 [7] та IEEE P1687 [8] (стандарт був затверджений як новий стандарт IEEE-SA Standards Board 3 листопада 2014 року [9]).

Існуючі рішення задач тестування та ремонту логічних компонентів цифрових систем поділяються на три класифікації:

1. Дублювання логічних елементів або областей кристалу, що приводить до подвоєння апаратної реалізації функціональності. При фіксації несправного елемента або області здійснюється перемикання за допомогою мультиплексора на справний компонент [10]. Запропоновані Xilinx - американським розробником та виробником інтегральних мікросхем програмованої логіки - моделі FPGA (Field Programmable Gate Array) також застосовуються під час ремонту компонентів FPGA від компанії Altera. При ремонті основною одиницею вимірювання є стовпчик або рядок.

2. Використання генетичних алгоритмів для діагностування та відтворення працездатності на основі автономної реконфігурації кристалу FPGA без використання зовнішніх пристроїв керування. Надійність діагностування дефектів дорівнює 99%, час ремонту - 36 мілісекунд замість 660 секунд, необхідних для стандартного конфігурування проекту [11].

3. Відновлення працездатності кристалів FPGA не критичних за часом за допомогою заміни локальних CLB (Complex Logic Block) на надлишкові запасні компоненти запропоновано в [12,13]. Доступний рівень об'єднання CLB, що підлягає заміні, для критично важливих додатків складає приблизно тисячі логічних елементів.

**Метою** дослідження є аналіз існуючих засобів та методів пошуку дефектів у цифрових системах на кристалах, визначення їх головних недоліків та розроблення алгоритму методу тестування, що буде досконаліший за таких попередників, наприклад, як послідовний метод діагностики несправностей, умовний алгоритм пошуку дефектів за допомогою побудови дерева, автоматичні системи тестування фірм Logic Vision, Synopsys, Mentor Graphics, Candence.

**Дедуктивно-паралельне моделювання несправності цифрових систем.** Модель дедуктивно-паралельного синхронного аналізу несправностей дискретного об'єкту дозволяє вирахувати на двійковому тест-векторі всі дефекти за одну ітерацію обробки схеми.

$$L = T \oplus F, \quad (1.1)$$

де -  $F = (F_{m+1}, F_{m+2}, \dots, F_i, \dots, F_n) (i = \overline{m+1, n})$  - сукупність функцій справної поведінки приладу;

$m$  - число входів;  $Y_i = F_i(X_{i1}, \dots, X_{ij}, \dots, X_{in_i})$  -  $n_i$ - вхідний  $i$ -й елемент схеми, що реалізує  $F_i$  для виявлення стану виходу  $Y_i$  на тест-векторі  $T_t$ ;  $X_{ij}$  -  $j$ -й вхід  $i$ -го елемента;

тест  $T = (T_1, T_2, \dots, T_t, \dots, T_k)$  - впорядкована сукупність двійкових векторів, доведена в процесі справного моделювання на множині вхідних, внутрішніх та вихідних ліній (виходів), об'єднана в матрицю

$$T = [T_{ti}] = \begin{bmatrix} T_{11} & T_{12} & \dots & T_{1i} & \dots & T_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ T_{t1} & T_{t2} & \dots & T_{ti} & \dots & T_{tn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ T_{k1} & T_{k2} & \dots & T_{ki} & \dots & T_{kn} \end{bmatrix} \quad (1.2)$$

невхідна координата якої визначається моделюванням функції:  $T_{ti} = Y_i = F_i(X_{i1}, \dots, X_{ij}, \dots, X_{in_i})$  на тест-векторі  $T_t$ ;  $L = (L_1, L_2, \dots, L_t, \dots, L_k)$  - множина дедуктивних схем або моделей, що визначаються виразом (1.1), де  $L_t = (L_{t1}, L_{t2}, \dots, L_{ti}, \dots, L_{tn})$ ;

$$L_{ti} = T_i \oplus F_i, \quad (1.3)$$

- дедуктивна функція (ДФ) паралельного моделювання несправностей на тест-векторі  $T_t$ , яка відповідає справному елементу  $F_i$ , що дає можливість вирахувати список вхідних несправностей, що транспортуються на вихід елементів  $F_i$  [14].

Синхронність моделі (1.1) може бути виявлена умовою:  $\Delta t = (t_{j+1} - t_j) \gg \tau \gg \tau_i$ , коли інтервал часу між зміною вхідних наборів  $(t_{j+1} - t_j)$ , що подаються на схему, набагато більше максимальної затримки схеми елементу. Така методика використовується в технологіях моделювання та дозволяє виключити час як несуттєвий параметр [14].

У загальному випадку, коли функція елементу задана таблицею істинності, використання формули (1.1) дозволяє отримати таблицю транспортування несправностей для тест-вектору  $T_t$ , завдяки якій можливо записати ДФ моделювання дефектів.

Наприклад, на рисунку 1 перший доданок - тест-вектор, другий та третій - таблиці істинності та транспортування дефектів.

$$\begin{array}{|c|c|c|} \hline X_1 & X_2 & Y_1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y_1 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y_1 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ \hline \end{array}$$

$$L_1 = X_1 X_2 \cup X_1 \bar{X}_2$$
  

$$\begin{array}{|c|c|c|} \hline X_1 & X_2 & Y_1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y_1 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y_1 \\ \hline 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ \hline \end{array}$$

$$L_2 = X_1 X_2 \cup X_1 \bar{X}_2 \cup \bar{X}_1 X_2$$

Рис. 1 – ДФ моделювання дефектів

В даному випадку дедуктивні функції  $L_1$ ,  $L_2$  записані у вигляді диз'юнктивної нормальної форми (ДНФ) по конститuentам одиниці таблиць транспортування дефектів.

Враховуючи розділення тесту на складові вектори рівняння (1.1), отримання ДФ для  $T_t \in T$  приймає наступний вигляд:  $L_t = T_t \oplus F$ .

Якщо функціональний опис цифрового приладу представлений примітивами, що формують стан усіх ліній схеми, то наступний вираз виступає у якості формули перетворення справної моделі примітива  $F_i$  на тест-векторі  $T_i$  в дедуктивну функцію  $L_{ti}$ :

$$L_{ti} = T_t \oplus F_i = f_{ti}[(X_{i1} \oplus T_{t1}), (X_{i2} \oplus T_{t2}), \dots, (X_{ij} \oplus T_{tj}), \dots, (X_{in_i} \oplus T_{tn_i})] \oplus T_{ti} \quad (1.4)$$

Даний вираз є основою дедуктивного аналізу цифрових проектів [15, 16].

Практична реалізація виразу (1.4) зводиться до наступного алгоритму:

1. Визначення для тест-вектора  $t = 0$ , що обробляється, початкового значення індексу та ініціалізація векторів перевірених дефектів:

$$\forall i (D_i^0 = 0; D_i^1 = 0) \quad (1.5)$$

2. Визначення номеру наступного вхідного набору  $t = t+1$  для  $T_t \in T$ . Якщо  $t > k$  (немає вхідних наборів), то кінець моделювання.

3. Справне моделювання всіх примітивів  $F_i (i = \overline{1, n})$  схеми на вхідному наборі  $T_t^X \in T_t$  з метою до визначення невхідних координат вектору  $T_t^{\bar{X}} \in T_t$ :

$$T_t^{\bar{X}} = f(T_t^X, F) \quad (1.6)$$

4. У відповідності до виразу 1.5 ініціалізація матриці дефектів, що перевіряються на тест-векторі  $M = [M_{ij}]$ . Ініціалізація вектору дефектів, що перевіряються,  $\forall i (S_i^0 = 0; S_i^1 = 0)$ . На основі

використання формули 1.4 реконфігурація всіх примітивів  $F_i (i = \overline{1, n})$  з метою отримання дедуктивної схеми  $L_t \leftarrow \forall i (L_{ti} = T_i \oplus F_i)$ .

5. Паралельне моделювання несправностей з метою до визначення координат відповідних нехвдним лініям схеми шляхом виконання реєстрових операцій над рядками матриці дефектів  $M$ , що перевіряється, за допомогою отриманих дедуктивних функцій.

6. Формування векторів несправностей, що перевіряються, до всіх входів матриці відповідним вихідним лініям схеми за допомогою формули:

$$S^0 = \left( \bigcup_{\forall i \in Y} M_i \right) \cap T_t; S^1 = \left( \bigcup_{\forall i \in Y} M_i \right) \cap \bar{T}_t \quad (1.7)$$

7. Якість тест-вектора  $T_t \in T$  при ідентичності списків несправностей у двох сусідніх ітераціях  $(S^0 \cup S^1)^{r-1} = (S^0 \cup S^1)^r$  (де  $r$  - це індекс ітерації) визначається формулою:

$$Q(T_t) = \frac{1}{2n} [\sum_{i=1}^n (S_i^0 + S_i^1)] \quad (1.8)$$

та здійснюється перехід до наступного пункту. Інакше виконується виключення дефектів із процесу моделювання за правилом:

$$(S_i^0 = S_i^1 = 0) \leftarrow \forall i [(S_i^0 \cup S_i^1 = 1)^{r-1} \& (S_i^0 \cup S_i^1 = 0)^r] \quad (1.9)$$

якщо спостерігається зникнення несправностей, що перевіряються в ітерації  $r$  порівняно з  $r-1$ :

$$\exists i [(S_i^0 \cup S_i^1 = 1)^{r-1} \& (S_i^0 \cup S_i^1 = 0)^r] \quad (1.10)$$

перехід до пункту п'ять.

8. Формування векторів перевірених несправностей відповідно до виразу

$$D^0 = D^0 \cup S^0, D^1 = D^1 \cup S^1 \quad (1.11)$$

та визначення якості тесту за формулою:

$$Q(T_t) = \frac{1}{2n} [\sum_{i=1}^n (D_i^0 + D_i^1)] \quad (1.12)$$

На рисунку 2 представлений універсальний функціональний примітив (симулятор) дедуктивно-паралельного аналізу несправностей.

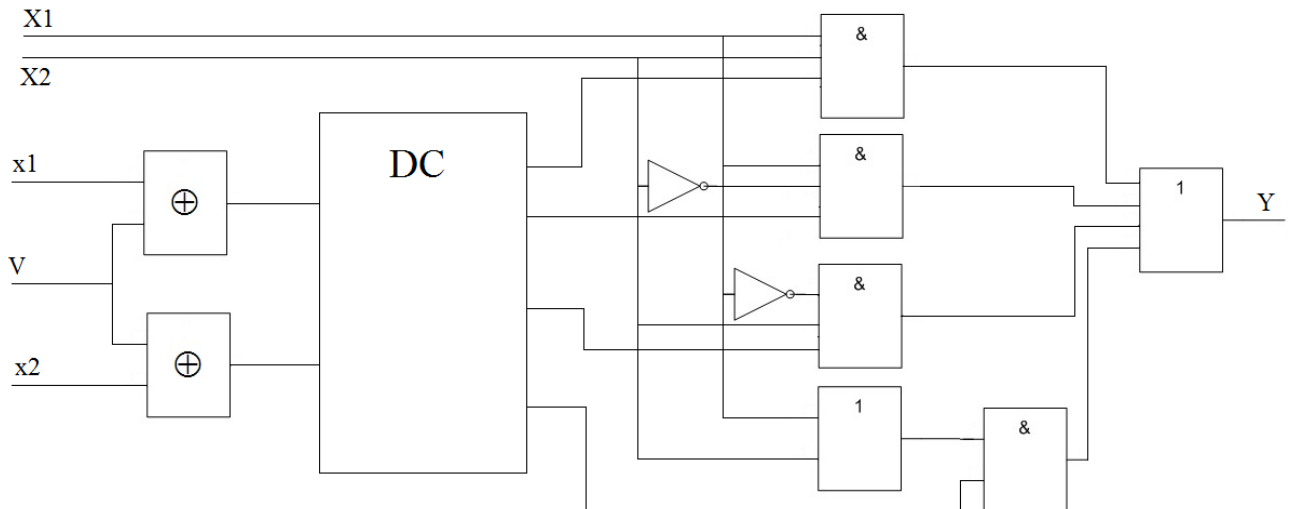


Рис. 2 – Симулятор несправних примітивів

У симуляторі представлені регістри ( $X_1, X_2$ ) та булеві ( $x_1, x_2$ ) входи, логічні функції (елементи І та АБО) вихідна регістрова змінна  $Y$ .

Таблиця 1 ілюструє паралельне моделювання вхідних 4-розрядних векторів несправностей для отримання на виході  $Y$  множини дефектів для логічних елементів, що перевіряються.

Таблиця 1 – Паралельне моделювання вхідних 4-розрядних векторів несправностей

I/АБО	x1x2	X1	X2	Y
0	00	0111	1011	0011
1	00	0111	1011	1111
0	11	0101	0110	0111
1	11	1101	0111	0101

Даний симулятор дає можливість трансформувати вентиляну модель  $F$  справної поведінки схеми у дедуктивну модель  $L$ , що не передбачає в процесі моделювання використання вентиляної моделі.

Як апаратна модель ДФ симулятор орієнтований на створення вбудованих засобів дедуктивно-паралельного моделювання, що в порівнянні з програмною реалізацією підвищує швидкодію аналізу в 10-1000 разів.

Модель дедуктивно-паралельного аналізу представлена на рисунку 3.

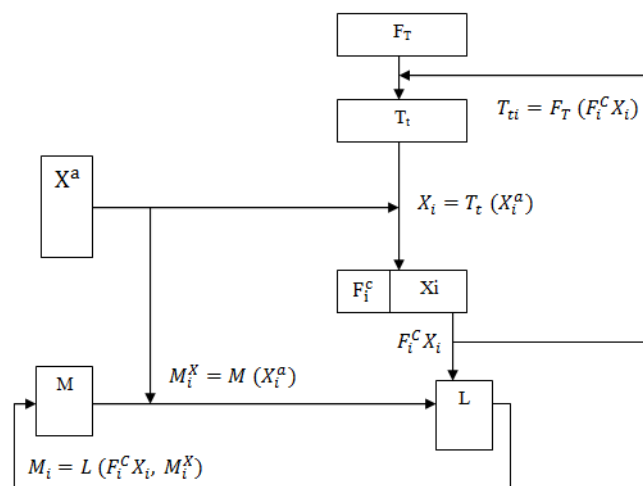


Рис. 3 – Модель дедуктивно-паралельного аналізу

де  $M$  і  $L$  - два додаткових модулі;  $Y_i$ - вихід логічного елемента;  $F_i^c$  - двійковий код елемента або його ідентифікатор;  $F_T$  - таблиця істинності сукупності булевих функцій для опису елементів схеми;  $(F_i^c * X_i)$  - вхідний набір;  $M_i = L(F_i^c X_i, M_i^X)$  - отриманий з  $F$  аналітичний вираз для визначення векторів несправностей, що перевіряються, об'єднаних в матрицю  $M$  за допомогою набору дедуктивних функцій  $L$ .

Дедуктивно-паралельний аналіз дефектів орієнтований на обробку великих проектів вентиляного та регістрового рівня опису з метою отримання таблиці несправностей та оцінки якості покриття тестом дефектів заданого класу. Об'єкт тестування представлений у формі структур, таблиць, булевих рівнянь, кубічних покриттів та реалізує складну цифрову систему, вбудовану в кристали SoC.

В даному випадку технологічно розглядати процедури синтезу на основі аналітичної форми. При цьому логічний елемент з двома вхідними лініями перетворюється на елемент з чотирма вхідними лініями, де  $x_1, x_2$  - булеві входи, а  $a$  і  $b$  - додаткові регістри. Регістрові входи служать для транспортування списків несправностей, а булеві (керуючі) - для виконання операцій над зовнішніми списками дефектів.

В таблиці 2 представлений синтез логічного елементу АБО.

Таблиця 2 – Синтез логічного елементу АБО

	(x,y)\(a,b)	00	01	11	10
$L = f(x,y,a,b) =$	00	0	1	1	1
	01	0	1	0	0
	11	0	0	1	0
	10	0	0	0	1

Мінімізація примітиву з таблиці 2 приводить до формування трьох дедуктивних функцій різної складності розв'язання.

$$\begin{aligned}
 1) L = f(x, y, a, b) &= (\bar{x}\bar{y} \cap ab) \cup (y \cap a\bar{b}) \cup (x \cap \bar{a}b) \cup (xy \cap b) \cup (xy \cap a) = \\
 &= (\bar{x}\bar{y} \cap ab) \cup (x \cap \bar{a}b) \cup (y \cap a\bar{b}) \cup [(xy \cap (a \cup b))] \\
 2) L = f(x, y, a, b) &= (\bar{x}\bar{y} \cap ab) \cup (y \cap a\bar{b}) \cup (x \cap \bar{a}b) \cup (xy \cap a) = \\
 &= (\bar{x}\bar{y} \cap ab) \cup (x \cap \bar{a}b) \cup [(ya \cap (x \cup \bar{b}))] \\
 3) L = f(x, y, a, b) &= (\bar{x}\bar{y} \cap ab) \cup (y \cap a\bar{b}) \cup (x \cap \bar{a}b) \cup (xy \cap b) \cup (xy \cap a) = \\
 &= (\bar{x}\bar{y} \cap ab) \cup [(ya \cap (x \cup \bar{b}))] \cup [(xb \cap (y \cup \bar{a}))].
 \end{aligned}
 \tag{1.13}$$

Результати мінімальних по Квайну функцій апаратно реалізуються у наступні схеми: на рисунку 4 представлена схема дедуктивного примітиву елементу АБО, на рисунку 5 - елементу І, а на рисунку 6 - елементу виключне АБО (операція за модулем 2 - XOR).

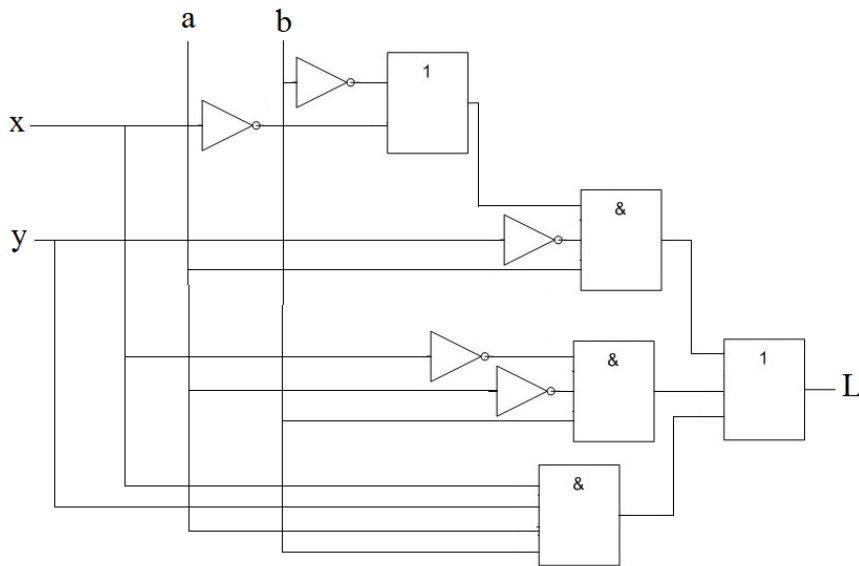


Рис. 4 – Схема дедуктивного примітиву елементу АБО

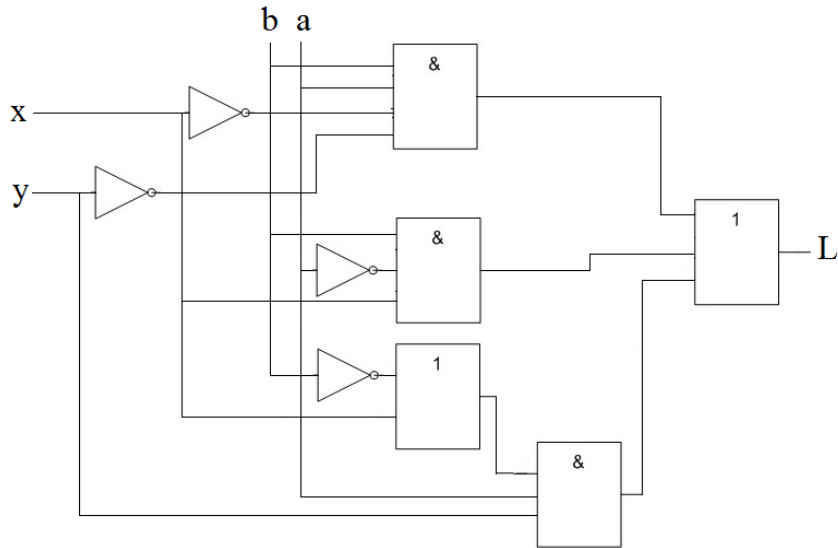


Рис. 5 – Схема дедуктивного примітиву елементу I

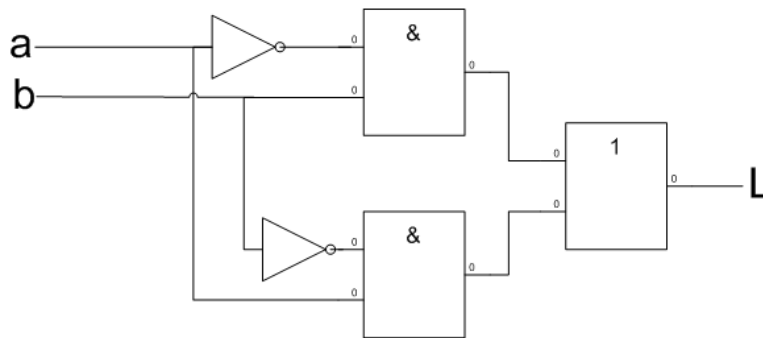


Рис. 6 – Схема дедуктивного примітиву елементу XOR

Регістровий рівень опису компонентів цифрової системи відрізняється функціональною складністю, що впливає на розмір таблиць істинності та кубічних покриттів. На цьому рівні розглядаються такі функціональності, як тригери, засувки, лічильники, регістри, мультиплексори, шинні структури.

Наприклад, синтез дедуктивної функції D-тригера з трьома булевими та трьома регістровими вхідними лініям  $Q = DC \cup \bar{C}\bar{D}Q(t-1)$ , функція якого представлена картою Карно в таблиці 3.

Таблиця 3 – Синтез дедуктивної функції D-тригера

(T)\(X)	000	001	011	010	110	111	101	100
000	0	1	1	0	1	1	0	0
001	0	1	1	0	0	0	1	1
011	0	1	1	0	1	1	0	0
010	0	1	1	0	0	0	1	1
110	0	0	1	1	1	0	0	1
111	0	0	1	1	0	1	1	0
101	0	0	1	1	1	0	0	1
100	0	0	1	1	0	1	1	0

$$\begin{aligned}
 L = f(c, d, q, C, D, Q) = & (\bar{c} \cap \bar{C}\bar{D}Q) \cup (\bar{C}\bar{D}Q) \cup (c \cap \bar{C}D\bar{Q}) \cup \\
 & \cup (\bar{c}\bar{d}\bar{q} \cap CD) \cup (\bar{c}\bar{d}\bar{q} \cap C\bar{D}) \cup (\bar{c}d\bar{q} \cap CD) \cup (\bar{c}\bar{d}\bar{q} \cap C\bar{D}) \\
 & \cup (cd\bar{q} \cap C\bar{Q}) \cup (cdq \cap CQ) \cup (c\bar{d}\bar{q} \cap C\bar{Q}) \cup (c\bar{d}\bar{q} \cap CQ)
 \end{aligned}
 \tag{1.14}$$



**Висновки.** Проведений аналіз існуючих засобів та методів пошуку дефектів у цифрових системах на кристалах та їх класифікація за трьома основними ознаками. Запропонований дедуктивний аналіз дефектів, що об'єднує в собі переваги дедуктивного виявлення списків несправностей, ефективного з точки зору математики, та виконання паралельних процедур, що орієнтовані на високошвидкісну обробку цифрових приладів вентильного, системного та реєстрового рівня опису.

#### Список бібліографічного опису

1. Ziad Noun. Wireless Approach for SIP and SOC Testing. Micro and nanotechnologies/Microelectronics. Université Montpellier II - Sciences et Techniques du Languedoc. English. URL: [https://theses.hal.science/file/index/docid/512832/filename/Ziad\\_Noun\\_-\\_Wireless\\_Approach\\_for\\_SiP\\_and\\_SoC\\_testing.pdf](https://theses.hal.science/file/index/docid/512832/filename/Ziad_Noun_-_Wireless_Approach_for_SiP_and_SoC_testing.pdf) (Дата звернення 05.05.2023).
2. Литвинова Е.И. Технологии встроенного тестирования system-in-package. URL: <https://cyberleninka.ru/article/n/tehnologii-vstroennogo-testirovaniya-system-in-package> (Дата звернення 10.05.2023).
3. Каравай М.Ф. Алгоритм построения проверяющего теста для кратных неисправностей по структуре комбинационного устройства. URL: [http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=at&paperid=7778&option\\_lang=rus](http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=at&paperid=7778&option_lang=rus) (Дата звернення 15.05.2023).
4. Олиховский С. Й., Белова М. М., Кочелаб Е. В. Кинетика образования и роста микродефектов в кристаллах. URL: <https://www.researchgate.net/..Kinetics-of-nucleation-and-growth-of-microdefects-in-crystals-in-Russian.pdf> (Дата звернення 15.05.2023).
5. Appello D., Bernardi P., Grosso M., Reorda M.S. System-in-package testing: problems and solutions // IEEE Design & Test of Computers. Vol. 23, Issue 3. May-June, 2021. P. 203 – 211.
6. Городецкий Ами Встроенные инструменты тестирования. URL: <https://cyberleninka.ru/article/n/vstroennye-instrumenty-testirovaniya> (Дата звернення: 23.05.2023).
7. Городецкий Ами Новый JTAG-стандарт IEEE 1149.7. URL: <https://cyberleninka.ru/article/n/novyy-jtag-standart-ieee-1149-7> (Дата звернення: 23.05.2023).
- 8 Farrokh Ghani Zadegan, Urban Ingelsson , Gunnar Carlsson and Erik Larsson. Design Automation for IEEE P1687. URL: [https://www.ida.liu.se/labs/eslab/publications/pap/db/farza\\_DATE11.pdf](https://www.ida.liu.se/labs/eslab/publications/pap/db/farza_DATE11.pdf) (Дата звернення 15.05.2023).
9. 1687 Proposed Hardware Architecture. URL: [https://grouper.ieee.org/groups/1687/HW\\_Post\\_Vote\\_r7\\_BW.pdf](https://grouper.ieee.org/groups/1687/HW_Post_Vote_r7_BW.pdf) . (Дата звернення 15.05.2023).
10. Subhashish M.S, Huang W.-J., Saxena N.R., Yu S.-Y., McCluskey E.J. Reconfigurable architecture for autonomous self-repair // IEEE Design & Test of Computers. - Volume 21, Issue 3. - May, 2019. -P.228-240.
11. Ross R., Hall R. A FPGA Simulation Using Asexual Genetic Algorithms for Integrated Self-Repair // Adaptive Hardware and Systems, 2018. -AHS 2018 - First NASA/ESA Conference on Volume.- Issue 15-18 June 2018. - P. 301-304
12. Habermann S., Kothe R., Vierhaus H.T. Built-in self repair by reconfiguration of FPGAs // Proceeding of the 12th IEEE International Symposium on On-Line Testing. - 2020. - P. 187-188.
13. Pontarelli, M. Ottavi, V. Vankamamidi, A. Salsano, F Lombardi. Reliability Evaluation of Repairable/Reconfigurable FPGAs // 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'21)/- October, 2021. - P. 227-235.
14. Hahanov V.I., Babich A.V., Hyduke S.M. Test Generation and Fault Simulation Methods on the Basis of Cubic Algebra for Digital Devices // Proceeding of the Euromicro Symposium on Digital System Design DSD2010/ - Warsaw, Poland. - September, 4-6, 2010. - P. 228-235.
15. Abramovici M., Breuer M.A. and Friedman A.D. Digital systems testing and testable design. - Computer Science Press. - 2022. - 652 p.

#### References

1. Ziad Noun. Wireless Approach for SIP and SOC Testing. Micro and nanotechnologies/Microelectronics. Université Montpellier II - Sciences et Techniques du Languedoc. English. URL: [https://theses.hal.science/file/index/docid/512832/filename/Ziad\\_Noun\\_-\\_Wireless\\_Approach\\_for\\_SiP\\_and\\_SoC\\_testing.pdf](https://theses.hal.science/file/index/docid/512832/filename/Ziad_Noun_-_Wireless_Approach_for_SiP_and_SoC_testing.pdf) (Accessed 05/05/2023).
2. Litvynova E.I. Technologies of built-in system-in-package testing. URL: <https://cyberleninka.ru/article/n/tehnologii-vstroennogo-testirovaniya-system-in-package> (Access date 05/10/2023).
3. Karavai M.F. Algorithm for building a verification test for multiple malfunctions based on the structure of a combinational device. URL: [http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=at&paperid=7778&option\\_lang=rus](http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=at&paperid=7778&option_lang=rus) (Access date 05/15/2023).
4. Olikhovskiy S.Y., Belova M.M., Kochelab E.V. Kinetics of formation and growth of microdefects in crystals. URL: <https://www.researchgate.net/..Kinetics-of-nucleation-and-growth-of-microdefects-in-crystals-in-Russian.pdf> (Accessed 05/15/2023).
5. Appello D., Bernardi P., Grosso M., Reorda M.S. System-in-package testing: problems and solutions // IEEE Design & Test of Computers. Vol. 23, Issue 3. May-June, 2021. P. 203-211.



6. Horodetsky Ami Built-in testing tools. URL: <https://cyberleninka.ru/article/n/vstroennyye-instrumenty-testirovaniya> (Access date: 05/23/2023).
7. Horodetsky Ami New JTAG standard IEEE 1149. URL: <https://cyberleninka.ru/article/n/novyy-jtag-standart-ieee-1149-7> (Access date: 05/23/2023).
- 8 Farrokh Ghani Zadegan, Urban Ingelsson, Gunnar Carlsson and Erik Larsson. Design Automation for IEEE P1687. URL: [https://www.ida.liu.se/labs/eslab/publications/pap/db/farza\\_DATE11.pdf](https://www.ida.liu.se/labs/eslab/publications/pap/db/farza_DATE11.pdf) (Date accessed 05/15/2023).
9. 1687 Proposed Hardware Architecture. URL: [https://grouper.ieee.org/groups/1687/HW\\_Post\\_Vote\\_r7\\_BW.pdf](https://grouper.ieee.org/groups/1687/HW_Post_Vote_r7_BW.pdf) . (Date of application 05/15/2023).
10. Subhasish M.S., Huang W.-J., Saxena N.R., Yu S.-Y., McCluskey E.J. Reconfigurable architecture for autonomous self-repair // IEEE Design & Test of Computers. - Volume 21, Issue 3. - May, 2019. - P.228-240.
11. Ross R., Hall R. A FPGA Simulation Using Asexual Genetic Algorithms for Integrated Self-Repair // Adaptive Hardware and Systems, 2018. -AHS 2018 - First NASA/ESA Conference on Volume.- Issue 15-18 June 2018. - P. 301-304
12. Habermann S., Kothe R., Vierhaus H.T. Built-in self repair by reconfiguration of FPGAs // Proceeding of the 12th IEEE International Symposium on On-Line Testing. - 2020. - P. 187-188.
13. Pontarelli, M. Ottavi, V. Vankamamidi, A. Salsano, F. Lombardi. Reliability Evaluation of Repairable/Reconfigurable FPGAs // 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'21)/- October, 2021. - P. 227-235.
14. Hahanov V.I., Babich A.V., Hyduke S.M. Test Generation and Fault Simulation Methods on the Basis of Cubic Algebra for Digital Devices // Proceeding of the Euromicro Symposium on Digital System Design DSD2010/ - Warsaw, Poland. - September, 4-6, 2010. - P. 228-235.
15. Abramovici M., Breuer M.A. and Friedman A.D. Digital systems testing and testable design. - Computer Science Press. - 2022. - 652 p.