

УДК 681.326.7

к.т.н. Успенський О.А. <http://orcid.org/0000-0001-6953-421X> (ВІТІ)к.т.н. Куліков В.М. <http://orcid.org/0000-0002-1015-5802> (ІС33І)к.т.н. Рябцев В.В. <http://orcid.org/0000-0001-8331-0132> (ІС33І)Мітін С.В. <http://orcid.org/0000-0002-4936-2569> (ІС33І)

МЕТОДИКА ПОБУДОВИ ПОВНИХ ПЕРЕВІРЯЮЧИХ ТЕСТІВ ДЛЯ ЦИФРОВИХ ПРИСТРОЇВ

У статті розглядається методика побудови повного тесту для перевірки цифрових пристроїв. В якості базового наведено загальний алгоритм побудови докладного тесту відносно несправностей довільної кратності. Опис методики викладено відносно одиночних константних помилок у цифрових комбінаційних схемах. Розглядаються несправності, які не перетворюють схему з категорії цифрових схем і спотворюють лише функції логічних елементів. До таких несправностей віднесено помилки типу закріплення значень сигналів в 0 та 1 на входах і виходах схеми та на входах і виходах логічних елементів. Показано, що повний тест для перевірки одиночних константних несправностей викриває в цифровій схемі також більшість несправностей кратності 2 і вище. Методика базується на застосуванні методів побудови тесту для пари технічних станів цифрової схеми і методів моделювання цифрових схем з несправностями. Для розв'язку задачі побудови тесту для пари станів схеми запропоновано використання методу, заснованого на пошуку термінальної вершини в дереві призначення сигналів і скороченні перебору методом фокусованого пошуку. Моделювання схем з несправностями в методиці виконується дедуктивним методом. Скорочення перебору методом фокусованого пошуку і заміна в дедуктивному методі теоретико-множинних операцій на побітові операції диз'юнкції і кон'юнкції робочих полів дозволяє значно зменшити час на отримання рішення при практичному застосуванні методики. В якості прикладу розглядається покрокова процедура побудови докладного тесту для перевірки комбінаційної схеми C17 ISCAS. Правильність результатів побудови тесту на кожному кроці демонструють схеми, які відповідають справному стану і стану з несправністю. Запропонована методика може розглядатися в якості базової для застосування при створенні докладних тестів для цифрових схем з пам'яттю. Докладний тест, отриманий при використанні даної методики, може значно знизити складні рішення діагностичної задачі, тобто задачі локалізації місця виникнення несправності.

Ключові слова: цифровий пристрій, технічне діагностування, перевіряючий тест, несправності цифрового пристрою, кратність несправності, модель цифрового пристрою, методи побудови тестів.

Успенский А.А., Куликов В.М., Рябцев В.В., Митин С.В. Методика построения полных проверяющих тестов для цифровых устройств.

В статье рассматривается методика построения полного проверяющего теста для цифровых устройств. В качестве базового приведен общий алгоритм построения полного теста относительно несправностей произвольной кратности. Описание методики изложено относительно одиночных константных несправностей в цифровых комбинационных схемах. Рассматриваются несправности, которые не преобразуют схему из категории цифровых схем и искажают только функции логических элементов. К таким несправностям относятся несправности типа закрепления значений сигналов в 0 и 1 на входах и выходах схемы и входах и выходах логических элементов. Показано, что полный тест для проверки одиночных константных несправностей выявляет в цифровой схеме также большинство несправностей кратности 2 и выше. Методика базируется на применении методов построения теста для пары технических состояний цифровой схемы и методов моделирования цифровых схем с несправностями. Для решения задачи построения теста для пары состояний схемы предложено использование метода, основанного на поиске терминальной вершины в дереве назначения сигналов и сокращении перебора методом фокусированного поиска. Моделирование схем с несправностями в методике выполняется дедуктивным методом. Сокращение перебора методом фокусированного поиска и замена в дедуктивном методе теоретико-множественных операций на побитовые операции дизъюнкции и конъюнкции рабочих полей, позволяет значительно уменьшить время на получение решения при практическом применении методики. В качестве примера рассматривается пошаговая процедура построения полного проверяющего теста для комбинационной схемы C17 ISCAS. Правильность результатов построения теста на каждом шагу демонстрируется схемам, которые соответствуют исправному состоянию и состоянию с несправностью. Предложенная методика может рассматриваться в качестве базовой для применения при создании полных тестов для цифровых схем с памятью. Полный тест, полученный при использовании данной методики, может значительно снизить сложные решения диагностической задачи, то есть задачи локализации места возникновения несправности.

Ключевые слова: цифровое устройство, техническое диагностирование, проверяющий тест несправности цифрового устройства, кратность несправности, модель цифрового устройства, методы построения тестов.

O.Uspenskiy, V.Kulikov, V.Riabtsev, S.Mitin. Methodology for constructing a complete validation test for digital devices.

The article reviews the methodology for constructing a complete validation test for digital devices. The general algorithm for constructing a complete test with respect to faults of arbitrary multiplicity is given as a basic one. The description of the method is made with respect to single constant faults in digital combinational circuits. We consider faults which do not convert the circuit from the category of digital circuits and distort only the functions of logic elements. Such malfunctions include problems such as fixing signal values at 0 and 1 at the inputs and outputs of the circuit and at the inputs and outputs of logic elements. It is shown that a complete test for checking single constant faults in the digital circuit also detects most faults of multiplicity 2 and higher. The methodology is based on the application of methods for constructing a test for a pair of technical states of a digital circuit and methods for modelling digital circuits with faults. To solve the problem of constructing a test for a pair of circuit states, it is proposed to use a method based on finding a terminal node in the signal assignment tree and on reducing the enumeration by the focused search method. Modelling of circuits with faults in the technique is performed by the deductive method. Reduction of enumeration by the focused search method and replacement in the deductive method of set-theoretic operations by bitwise operations of disjunction and conjunction of working fields can significantly reduce the time to obtain a solution in the practical application of the method. As an example, the step-by-step procedure for constructing a complete test for the C17 ISCAS combinational circuit is shown. The correctness of the test results at each step is demonstrated by the diagrams which correspond to the good state with a fault. The proposed technique can be considered as a basic one for application when creating complete tests for digital circuits with memory. A complete test obtained with this technique can significantly reduce complex solutions to the diagnostic problem, that is the problem of localizing the location of a malfunction.

Keywords: digital device, technical diagnostics, checking test of digital device malfunction, multiplicity of malfunction, digital device model, test construction methods.

Постановка завдання в загальному вигляді.

З науки "Технічна діагностика" відомо, що проблема побудови повного тесту для перевірки 100 % несправностей цифрового пристрою (перевіряючого тесту), насамперед потребує вирішення задачі побудови набору вхідних сигналів, по реакції на яку можливо розрізнити пару технічних станів s_i та s_j даного пристрою. На сьогоднішній день розроблено достатньо велику кількість методів її вирішення. Найбільше значення для практики с точки зору повноти покриття множини несправностей за думкою авторів статті набули детерміновані та випадкові методи генерації тестів та їх комбінації. Випадкові методи характеризуються меншою складністю але не гарантують досягнення необхідної повноти тестів. Детерміновані методи хоча й мають більшу складність проте саме їм надається перевага для рішення задачі побудови тестів із заданою повнотою. Спробу викласти їх порівняльну характеристику надано у [1].

Задача побудови діагностичного тесту не має сьогодні цікавого для практики рішення. Локалізація несправності у цифровому пристрої може бути виконана із застосуванням повного перевіряючого тесту разом з дедуктивним моделюванням, на яке покладається зменшення кількості підозрюваних несправних станів.

В статті [2] розглянуто алгоритм побудови повного перевіряючого тесту для цифрового пристрою щодо заданої множини несправностей. Показано, що задача може бути вирішена за допомогою двох процедур: генератора тестів для обраної несправності і моделі пристрою, яка дозволяє визначати несправності, для яких тест ще не знайдений. В якості генератора запропоновано використовувати процедуру призначення сигналів і скорочення перебору методом фокусованого пошуку. Для моделювання схем з несправностями пропонується використовувати дедуктивний метод, викладений в статті [3].

Аналіз публікацій за темою дослідження. Проблема побудови тестових наборів сигналів із заданою повнотою виявлення несправностей набуває підвищеної актуальності при зростанні складності цифрового пристрою, тобто при збільшенні кількості логічних елементів (ветилів) у схемі що підлягає тестуванню. Таке становище змушує фахівців шукати нові підходи до проведення діагностування технічних об'єктів. Наприклад, у [4] показано, що коди з сумуванням можуть бути ефективно використані при побудові систем функціонального контролю комбінаційних логічних пристроїв і, особливо, при побудові систем з виявленням всіх одиночних несправностей в контрольованому пристрої. Проте даний підхід може забезпечити перевірку правильності функціонування і не вирішує проблеми локалізації місця виникнення помилки.

В [5] та деяких інших роботах пропонуються теорія і приклади реалізації кубітних моделей, методів і алгоритмів для підвищення швидкодії існуючих програмних і апаратних засобів аналізу цифрових обчислювальних пристроїв за рахунок збільшення розмірності структур даних і пам'яті для одночасного зберігання оброблюваних станів.

Досвід аналізу різних способів побудови систем діагностування цифрових пристроїв на надвеликих інтегральних схемах доводить ефективність систем вбудованого контролю і самотестування [6]. В пристроях, побудованих на програмованих логічних схемах, діагностичне забезпечення створюється на етапі розробки і записується на етапі виготовлення цього пристрою. При цьому можуть бути застосовані як випадкові, так і детерміновані методи побудови перевіряючих і діагностичних тестів.

Вбудовані системи діагностування і самоконтроля потребують використання якісних верифікаційних тестів. В даний час не вирішено проблему побудови тестової послідовності мінімальної довжини. Підхід до оцінки якості тестових послідовностей викладено у [7].

Метою статті є підвищення ефективності діагностування цифрових пристроїв за рахунок розробки методики застосування методів фокусованого пошуку та дедуктивного моделювання для вирішення задачі побудови повного первіряючого тесту. В статті наведено приклад побудови тесту для цифрової комбінаційної схеми С17.

Виклад основного матеріалу

У даній статті пропонується поглиблений і менш формальний розгляд алгоритму побудови повного перевіряючого тесту на прикладі простої комбінаційної схеми, відомої як схема С17 ISCAS-85. Далі наведено початковий опис схеми і її зображення у вигляді логічної мережі з п'ятьма входами (позначеними 1, 2, 3, 6 і 7) і шістьма елементами І-НЕ, два з яких (22-й і 23) формують вихідні сигнали схеми (Рис.1). Функції елементів наведені в диз'юнктивній нормальній формі (ДНФ).

# total number of lines in the netlist	17
# simplistically reduced equivalent fault set size =	22
# lines from primary input gates	5
# lines from primary output gates	2
# lines from interior gate outputs	4
# lines from ** 3 ** fanout stems	6
# avg_fanin = 2.00, max_fanin = 2	
# avg_fanout = 2.00, max_fanout = 2	

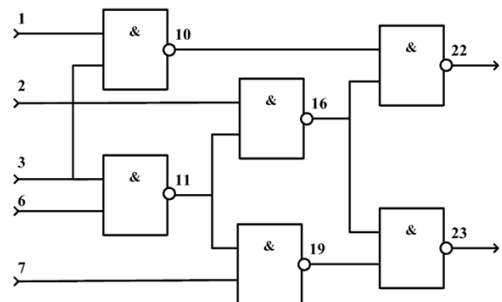


Рис.1. Схема С17

Функції елементів в ДНФ:

$$P_{10} = \overline{P_1 + P_3}; \quad \overline{P_{10}} = P_1 P_3;$$

$$P_{11} = \overline{P_3 + P_6}; \quad \overline{P_{11}} = P_3 P_6;$$

$$P_{16} = \overline{P_2 + P_{11}}; \quad \overline{P_{16}} = P_2 P_{11};$$

$$P_{19} = \overline{P_{11} + P_7}; \quad \overline{P_{19}} = P_{11} P_7;$$

$$P_{22} = \overline{P_{10} + P_{16}}; \quad \overline{P_{22}} = P_{10} P_{16};$$

$$P_{23} = \overline{P_{16} + P_{19}}; \quad \overline{P_{23}} = P_{16} P_{19};$$

де P_1, P_2, \dots, P_{23} – змінні числення висловлювань, значеннями яких можуть бути "True" або "False" в залежності від значень сигналів на виходах елементів і входах схеми 1, 2, 3, 6, 7. $p_k = \text{"True"}$, якщо істинним є висловлювання "Значення сигналу на виході елемента k дорівнює 1", в іншому випадку $\overline{p_k} = \text{"True"}$, тобто істинним є висловлювання "Значення сигналу на виході елемента k дорівнює 0" (k - номер елемента).

Представлення схеми функціями в ДНФ використовується при вирішенні задачі побудови тесту для пари технічних станів цієї схеми методом фокусованого пошуку [2].

Формування списку несправностей

В цифровій схемі, що розглядається, кількість можливих несправностей дорівнює 22 (simplistically reduced equivalent fault set size =22). При формуванні списку несправностей враховано наступне [8]:

- 1) несправності можуть виникати тільки на входах і виходах схеми і на входах і виходах логічних елементів;
- 2) розглядаються тільки константні несправності типу стійке закріплення сигналу в **0** (const 0) і стійке закріплення в **1** (const 1);
- 3) тест, побудований для виявлення в схемі тільки одиночних константних несправностей, виявляє і всі або, принаймі, більшість несправностей кратності 2 і вище;
- 4) інші несправності в реальних цифрових пристроях можуть бути представлені моделлю одиночних константних несправностей на входах і виходах схеми і логічних елементів.

Сказане розглядається як підстава для твердження про те, що тест, побудований для сформованого списку несправностей, є повним.

Список всіх одиночних константних несправностей містить кількість несправностей, що дорівнює подвоєній кількості входів і виходів схеми плюс подвоєне число входів і виходів елементів схеми. Мінімізація кількості несправностей виконується шляхом виключення з розгляду несправностей, які є нерозрізненними. Наприклад, закріплення сигналу в **0** на будь-якому вході елемента І-НЕ неможливо відрізнити від закріплення сигналу в **1** на виході цього елемента.

Несправності цифрового пристрою моделюються сигналами, що приймають сталі (незмінні) значення на відповідному вході/виході логічного елемента/схеми. Сигнали та несправності в даній статті позначаються наступним чином:

– сигнали/несправності на виходах логічних елементів та входах схеми – A^k , де A – номер елемента/входу схеми, k – значення сигналу або його сталі значення (несправність) на вході схеми або виході логічного елемента;

– сигнали/несправності входів логічних елементів – A^k_n , де A – номер логічного елемента, k – сталі значення (несправність) на вході логічного елемента, n – номер логічного елемента або входу схеми, з якого сигнал подається на вхід елемента A .

Таким чином до списку одиночних константних несправностей схеми С17 входять наступні несправності: $1^1, 2^1, 3^1, 10^1_3, 11^1_3, 3^0, 6^1, 7^1, 10^1, 11^1, 16^1_{11}, 19^1_{11}, 11^0, 16^1, 22^1_{16}, 23^1_{16}, 16^0, 19^1, 22^1, 22^0, 23^1, 23^0$.

Модель схеми С17 в присутності одиночних константних несправностей

Модель схеми може бути представлена у вигляді функцій елементів в ДНФ, в які додано змінні справних технічних станів схеми і несправностей. Технічні стани схеми будемо позначати змінними s_0, s_1, \dots, s_{33} . Для позначення справних технічних станів будемо дописувати в ідентифікатори змінних літеру n . Семантика змінних технічних станів для схеми на Рис.1 приведена в Таблиці 1. (Розгляд методу відбору несправностей виходить за рамки даної статті.

Таблиця 1

Технічні стани схеми С17

Змінна технічного стану	Номер елемента	Технічний стан	Несправність
s_{0n}		справний стан схеми	
s_{1n}	1	справний	
s_2	1	const 1	1^1
s_{3n}	2	справний	
s_4	2	const 1	2^1
s_{5n}	3	справний	
s_6	3	const 0	3^0
s_7	3	const 1	3^1

s_8	10	const 1	10^1_3
s_9	11	const 1	11^1_3
s_{10n}	6	справний	
s_{11}	6	const 1	6^1
s_{12n}	7	справний	
s_{13}	7	const 1	7^1
s_{14n}	10	справний	
s_{15}	10	const 1	10^1
s_{16n}	11	справний	
s_{17}	11	const 0	11^0
s_{18}	11	const 1	11^1
s_{19}	16	const 1	16^1_{11}
s_{20}	19	const 1	19^1_{11}
s_{21n}	16	справний	
s_{22}	16	const 0	16^0
s_{23}	16	const 1	16^1
s_{24}	22	const 1	22^1_{16}
s_{25}	23	const 1	23^1_{16}
s_{26n}	19	справний	
s_{27}	19	const 1	19^1
s_{28n}	22	справний	
s_{29}	22	const 0	22^0
s_{30}	22	const 1	22^1
s_{31n}	23	справний	
s_{32}	23	const 0	23^0
s_{33}	23	const 1	23^1

З урахуванням позначень у Таблиці 1 можна сформувати модель схеми С17 в присутності одиночних константних несправностей. Наприклад, елемент 16 може перебувати у справному стані (технічний стан s_{21n}) або знаходитися в одному з своїх несправних технічних станах s_{19} , s_{22} , s_{23} . Функції, що реалізуються елементом 16 в цих станах, мають вигляд:

<p>- в стані s_{21n}</p> $P_{16} = s_{21n} \overline{P_2} + s_{21n} \overline{P_{11}}; \quad \overline{P_{16}} = s_{21n} P_2 P_{11};$ <p>- в стані s_{22}</p> $P_{16} = \text{"False"}; \quad \overline{P_{16}} = s_{22};$	<p>- в стані s_{19}</p> $P_{16} = s_{19} \overline{P_2}; \quad \overline{P_{16}} = s_{19} P_2;$ <p>- в стані s_{23}</p> $P_{16} = s_{23}; \quad \overline{P_{16}} = \text{"False"};$
--	--

Функції всіх несправностей елемента 16 утворюється простою диз'юнкцією функцій всіх станів:

$$\overline{P_{16}} = s_{21n} \overline{P_2} + s_{21n} \overline{P_{11}} + s_{19} \overline{P_2} + s_{23};$$

$$P_{16} = s_{21n} P_2 P_{11} + s_{19} P_2 + s_{22};$$

Аналогічно можуть бути сформовані функції несправностей всіх елементів схеми.

$$\overline{P_{10}} = s_{14n} \overline{P_3} + s_{14n} \overline{P_1} + s_8 \overline{P_1} + s_{15};$$

$$P_{10} = s_{14n} P_3 P_1 + s_8 P_1;$$

$$\overline{P_{11}} = s_{16n} \overline{P_6} + s_{16n} \overline{P_3} + s_9 \overline{P_6} + s_{18};$$

$$P_{11} = s_{16n} P_6 P_3 + s_9 P_6 + s_{17};$$

$$P_{19} = s_{26n} \overline{P_7} + s_{26n} \overline{P_{11}} + s_{20} \overline{P_7} + s_{27};$$

$$\begin{aligned} \overline{P_{19}} &= s_{26n} \overline{P_7} \overline{P_{11}} + s_{20} \overline{P_7}; \\ \overline{P_{22}} &= s_{28n} \overline{P_{16}} + s_{28n} \overline{P_{10}} + s_{24} \overline{P_{10}} + s_{30}; \\ \overline{P_{22}} &= s_{28n} \overline{P_{16}} \overline{P_{10}} + s_{24} \overline{P_{10}} + s_{29}; \\ \overline{P_{23}} &= s_{31n} \overline{P_{19}} + s_{31n} \overline{P_{16}} + s_{25} \overline{P_{19}} + s_{33}; \\ \overline{P_{23}} &= s_{31n} \overline{P_{19}} \overline{P_{16}} + s_{25} \overline{P_{19}} + s_{32}; \end{aligned}$$

Для отримання функцій несправностей входів схеми потрібно ввести додаткові вхідні змінні $\overline{P_{01}}, \overline{P_{02}}, \overline{P_{03}}, \overline{P_{06}}, \overline{P_{07}}$, значення яких залежать тільки від вхідних сигналів схеми, в той час як значення змінних $\overline{P_1}, \overline{P_2}, \overline{P_3}, \overline{P_6}, \overline{P_7}$ залежать від вхідних сигналів і технічних станів входів схеми:

$$\left. \begin{aligned} \overline{P_1} &= s_{1n} \overline{P_{01}} + s_2; \\ \overline{P_1} &= s_{1n} \overline{P_{01}}; \\ \overline{P_2} &= s_{3n} \overline{P_{02}} + s_4; \\ \overline{P_2} &= s_{3n} \overline{P_{02}}; \end{aligned} \right\} \begin{aligned} \overline{P_3} &= s_{5n} \overline{P_{03}} + s_7; \\ \overline{P_3} &= s_{5n} \overline{P_{03}} + s_6; \\ \overline{P_6} &= s_{10n} \overline{P_{06}} + s_{11}; \\ \overline{P_6} &= s_{10n} \overline{P_{06}}; \end{aligned} \right\} \begin{aligned} \overline{P_7} &= s_{12n} \overline{P_{07}} + s_{13}; \\ \overline{P_7} &= s_{12n} \overline{P_{07}}; \end{aligned}$$

Функції елементів в ДНФ, сформовані з додаванням в них змінних станів, використовуються для побудови шкал і робочих полів при пошуку несправностей, які викриваються або не викриваються знайденим тестом в дедуктивному методі моделювання [3].

Методика побудови повних перевіряючих тестів для цифрових схем

Методика побудови перевіряючої послідовності зводиться до рішення двох задач: побудови вхідної послідовності, що розрізняє пару станів s_i, s_j цифрової схеми, і визначення стану s_k , який не відрізняється по реакції схеми від справного стану на вже знайденій частині перевіряючої послідовності [9]. Зазначені задачі відомі в технічній діагностиці, як, відповідно, зворотна та пряма і вважаються класичними [10].

Алгоритм побудови тесту для перевірки у цифровій схемі несправностей, кратність яких не перевищує k , може бути представлений у формі послідовності рішень прямої і зворотної задач [11]. Алгоритм представлено на Рис. 2. Далі представлено його покроковий опис.

Крок 1. Встановити початкові значення для кратності (k) несправностей, тестової послідовності (X), несправності (S_k) кратності k : $k=1$, $X=\emptyset$, $S_k = \langle \text{будь-яка несправність кратності } 1 \rangle$. (Блок 1.)

Крок 2. Побудувати тест T_k для пари станів S_k, S_n , де S_n – справний стан. Якщо T_k не існує, то S_k видаляється з розгляду. Якщо T_k існує, то приєднати його до X : $X = X \parallel T_k$. (Блоки 2, 3, 4.)

Крок 3. Знайти стан S_k , який не відрізняється від S_n по реакції на X . Якщо S_k існує, то перейти до кроку 2, інакше перейти до кроку 4. (Блоки 5, 6.)

Крок 4. Змінити кратність k несправностей, що розглядаються: $k=k+1$. Якщо значення k перевищило встановлений ліміт, то кінець. Інакше, перейти до кроку 3. (Блоки 7, 8.)

Результатом виконання алгоритму є тест, який перевіряє в схемі 100% несправностей, кратність яких не перевищує k . Такий тест називається *повним перевіряючим тестом*.

В роботах [8,12] показано, що при збільшенні кратності несправностей, для яких вже побудовано тест X , кількість несправностей більшої кратності, які не викриваються тестом X , швидко зменшується. Тому для практики у більшості застосувань достатньою є гранична кратність несправностей, яка дорівнює 1.

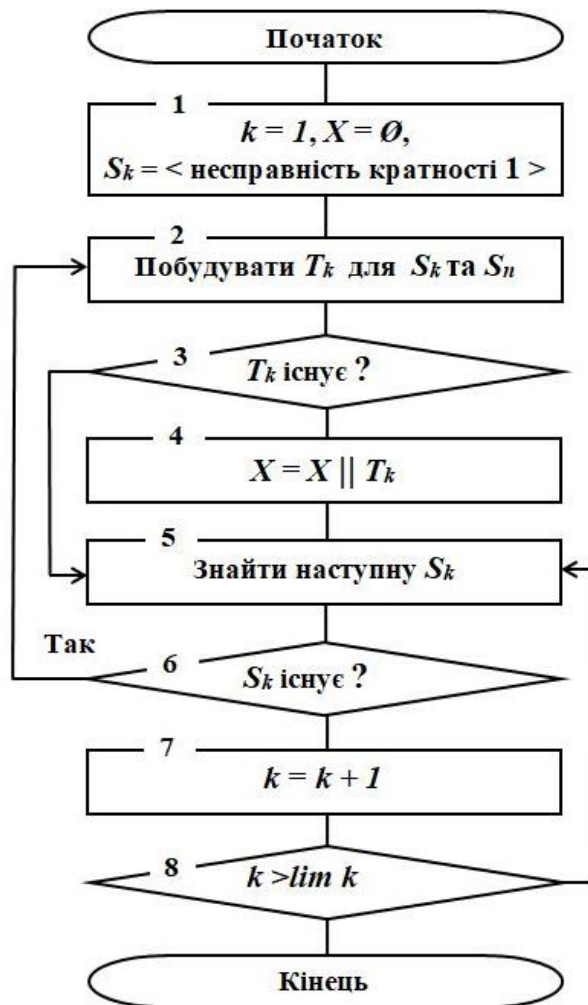


Рис.1. Алгоритм методики побудови повного перевіряючого тесту

Далі наведено покроковий опис ітеративної процедури побудови повного тесту для перевірки одиночних константних несправностей схеми С17. Побудова тесту для пари технічних станів схеми виконувалась методом фокусованого пошуку описаним в [2, 11]. Моделювання з метою визначення множини несправностей, які викриваються та не викриваються на вже побудованій частині повного тесту, здійснювалось дедуктивним методом наведеним в статті [3].

Опис реалізації методики супроводжується на кожній ітерації двома схемами С17 в справному стані і з черговою несправністю. На схемах продемонстровано, що побудований тест дійсно відрізняє справну схему від схеми у несправному стані. Результат застосування методики представлено у Таблиці 2.

Ітерація 1.

Крок 1.1. Встановити початкові значення для кратності (k) несправностей, тестової послідовності (X), несправності (S_k) кратності k : $k=1$, $X=\emptyset$, $S_1 = I^1$ (константна 1 на першому вході схеми).

Крок 1.2. Побудувати тест T_1 для пари станів S_1, S_n , де S_n – справний стан схеми. T_1 існує і дорівнює множені $\{1^0, 2^0, 3^1, 6^1, 7^1\}$. Додати T_1 до X : $X = X \parallel T_1 = \{1^0, 2^0, 3^1, 6^1, 7^1\}$.

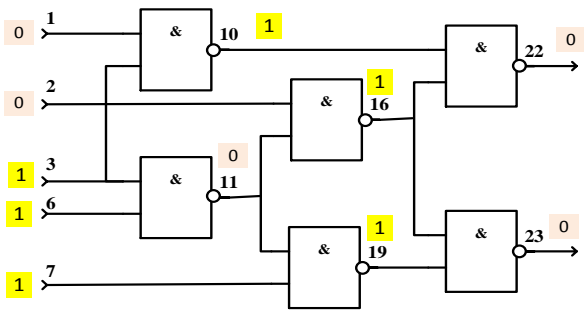


Схема С17

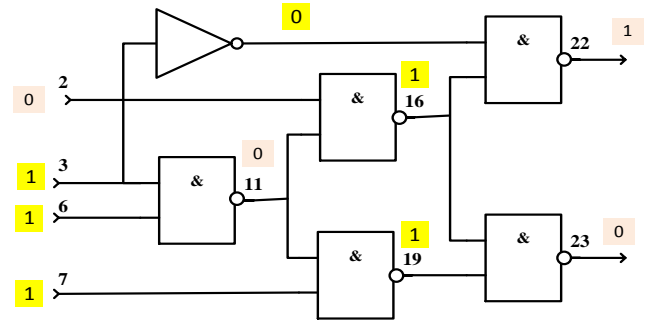


Схема С17 з несправністю I^1 (1 на вході 1 схеми)

Крок 1.3. Знайти несправність S_I , яка не відрізняється від S_n (справного стану) по реакції на X (тобто несправність, яка не викривається побудованим тестом). Така S_I існує: $S_I = 2^1$. Перейти до кроку 2 другої ітерації.

Ітерація 2.

Крок 2.2. Побудувати тест T_I для пари станів S_I, S_n , де $S_I = 2^1$, S_n – справний стан. T_I існує і дорівнює множені $\{1^0, 2^0, 3^0, 6^0, 7^0\}$. Додати T_I до X : $X = X // T_I = \{\{1^0, 2^0, 3^1, 6^1, 7^1\}, \{1^0, 2^0, 3^0, 6^0, 7^0\}\}$.

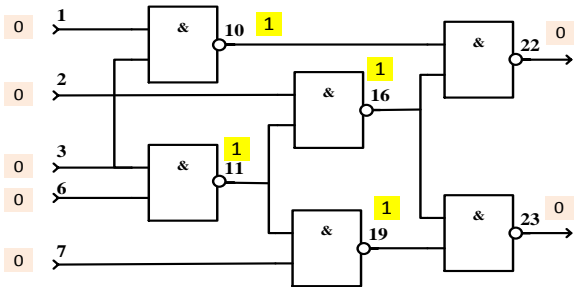


Схема С17

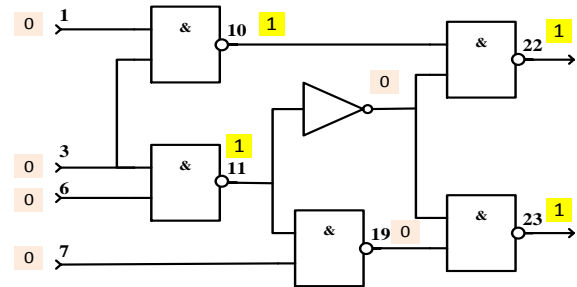


Схема С17 з несправністю 2^1 (1 на вході 2 схеми)

Крок 2.3. Знайти наступну несправність S_I , яка не відрізняється від S_n по реакції на X . Такою несправністю є несправність $S_I = 10^1_3$ (закріплення сигналу логічної одиниці на вхідній лінії елемента 10 від елемента 3). Перейти до кроку 2 третьої ітерації.

Ітерація 3.

Крок 3.2. Побудувати тест T_I для пари станів S_I, S_n , де $S_I = 10^1_3$, S_n – справний стан. T_I існує і дорівнює множені $\{1^1, 2^0, 3^0, 6^1, 7^0\}$. Додати T_I до X : $X = X // T_I = \{\{1^0, 2^0, 3^1, 6^1, 7^1\}, \{1^0, 2^0, 3^0, 6^0, 7^0\}, \{1^1, 2^0, 3^0, 6^1, 7^0\}\}$.

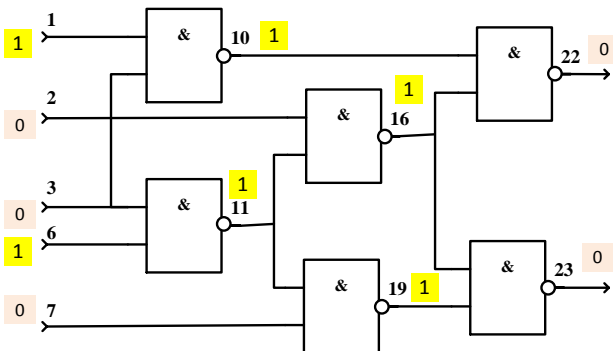


Схема С17

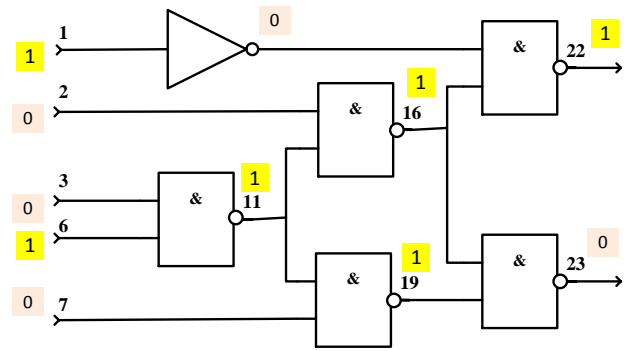


Схема С17 з несправністю 10^1_3 (1 на вході елемента 10)

Крок 3.3. Знайти наступну несправність S_I , яка не відрізняється від S_n по реакції на X . Такою несправністю є несправність $S_I = 11^1_3$ (закріплення сигналу логічної одиниці на вхідній лінії елемента 11 від елемента 3). Перейти до кроку 2 четвертої ітерації.

Ітерація 4.

Крок 4.2. Побудувати тест T_I для пари станів S_I, S_n , де $S_I = 11^1_3$, S_n – справний стан. T_I існує і дорівнює множені $\{1^0, 2^1, 3^0, 6^1, 7^0\}$. Додати T_I до X : $X = X // T_I = \{\{1^0, 2^0, 3^1, 6^1, 7^1\}, \{1^1, 2^0, 3^0, 6^0, 7^1\}, \{1^0, 2^0, 3^0, 6^0, 7^0\}, \{1^0, 2^1, 3^0, 6^1, 7^0\}\}$.

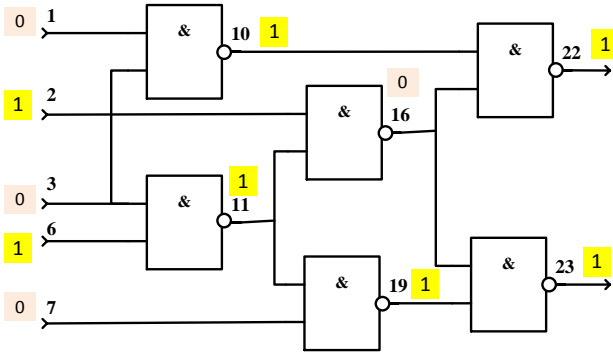


Схема С17

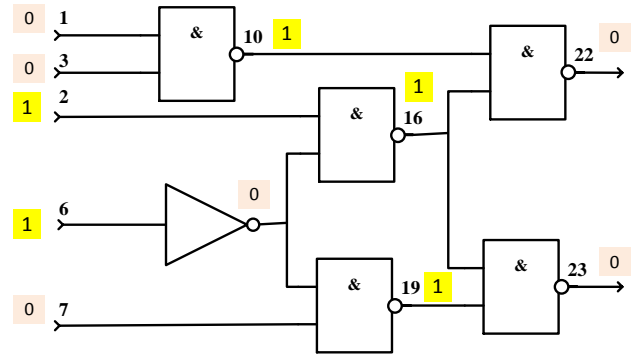


Схема С17 з несправністю 11^1_3 (1 на вході елемента 11)

Крок 4.3. Знайти наступну несправність S_I , яка не відрізняється від S_n по реакції на X . Такою несправністю є несправність $S_I = 6^1$ (закріплення сигналу логічної одиниці на виході 6 елемента). Перейти до кроку 2 п'ятої ітерації.

Ітерація 5.

Крок 5.2. Побудувати тест T_I для пари станів S_I, S_n , де $S_I = 6^1$, S_n – справний стан. T_I існує і дорівнює множені $\{1^0, 2^1, 3^1, 6^0, 7^0\}$. Додати T_I до X : $X = X // T_I = \{\{1^0, 2^0, 3^1, 6^1, 7^1\}, \{1^1, 2^0, 3^0, 6^0, 7^1\}, \{1^1, 2^0, 3^0, 6^1, 7^0\}, \{1^0, 2^0, 3^0, 6^0, 7^0\}, \{1^0, 2^1, 3^1, 6^0, 7^0\}\}$.

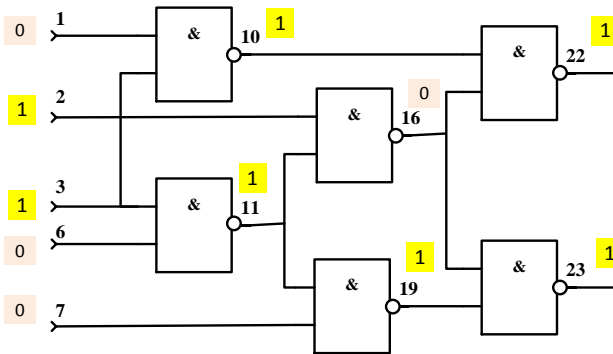


Схема С17

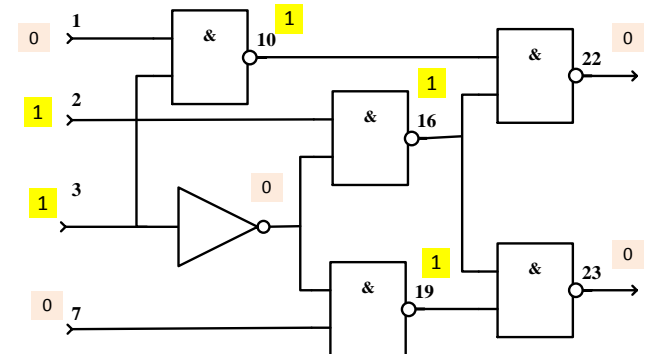


Схема С17 з несправністю 6^1 (1 на вході 6 схеми)

Крок 5.3. Знайти наступну несправність S_I , яка не відрізняється від S_n по реакції на X . Такою несправністю є несправність $S_I = 10^1$ (закріплення сигналу логічної одиниці на виході 10 елемента). Перейти до кроку 2 шостої ітерації.

Ітерація 6.

Крок 6.2. Побудувати тест T_I для пари станів S_I, S_n , де $S_I = 10^1$, S_n – справний стан. T_I існує і дорівнює множені $\{1^1, 2^0, 3^1, 6^0, 7^1\}$. Додати T_I до X : $X = X // T_I = \{\{1^0, 2^0, 3^1, 6^1, 7^1\}, \{1^1, 2^0, 3^0, 6^0, 7^1\}, \{1^1, 2^0, 3^0, 6^1, 7^0\}, \{1^0, 2^1, 3^0, 6^1, 7^0\}, \{1^0, 2^0, 3^0, 6^0, 7^0\}, \{1^1, 2^0, 3^1, 6^0, 7^1\}\}$.

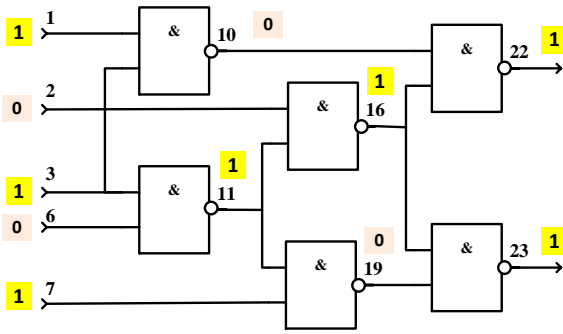


Схема С17

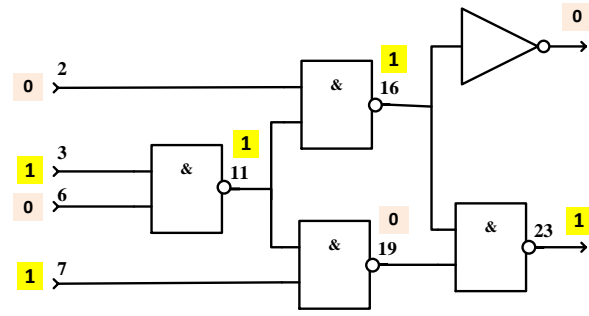


Схема С17 з несправністю 10^1 (1 на виході елемента 10)

Крок 6.3. Знайти наступну несправність S_I , яка не відрізняється від S_n по реакції на X . Такою несправністю є несправність $S_I = 16^1_{11}$ (закріплення сигналу логічної одиниці на вхідній лінії елемента 16 від елемента 11). Перейти до кроку 2 сьомої ітерації.

Ітерація 7.

Крок 7.2. Побудувати тест T_I для пари станів S_I, S_n , де $S_I = 16^1_{11}$, S_n – справний стан. T_I існує і дорівнює множенні $\{1^0, 2^1, 3^1, 6^1, 7^1\}$. Додати T_I до X : $X = X // T_I = \{\{1^0, 2^0, 3^1, 6^1, 7^1\}, \{1^1, 2^0, 3^0, 6^0, 7^1\}, \{1^1, 2^0, 3^0, 6^1, 7^0\}, \{1^0, 2^1, 3^0, 6^1, 7^0\}, \{1^0, 2^1, 3^1, 6^0, 7^0\}, \{1^0, 2^0, 3^0, 6^0, 7^0\}, \{1^0, 2^1, 3^1, 6^1, 7^1\}\}$.

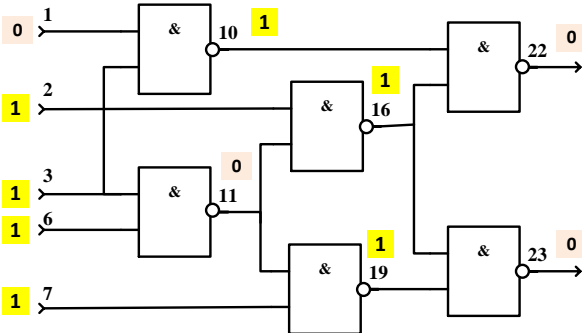


Схема С17

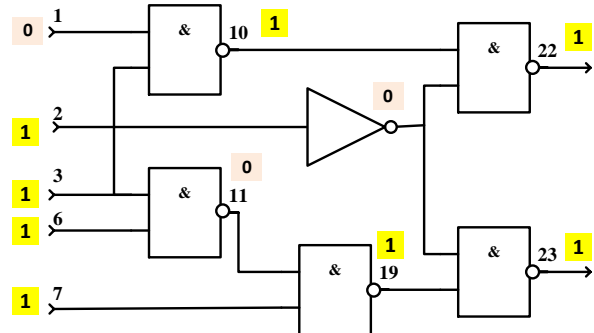


Схема С17 з несправністю 16^1_{11} (1 на вході елемента 16)

Крок 7.3. Знайти наступну несправність S_I , яка не відрізняється від S_n по реакції на X . Моделювання схеми показує, що тест X є повним перевіряючим тестом відносно заданої множини несправностей кратності 1. Перехід до кроку 4 не виконується.

$X = \{\{1^0, 2^0, 3^1, 6^1, 7^1\}, \{1^1, 2^0, 3^0, 6^0, 7^1\}, \{1^1, 2^0, 3^0, 6^1, 7^0\}, \{1^0, 2^1, 3^0, 6^1, 7^0\}, \{1^0, 2^1, 3^1, 6^0, 7^0\}, \{1^0, 2^0, 3^0, 6^0, 7^0\}, \{1^0, 2^1, 3^1, 6^1, 7^1\}\}$

Таблиця 2

Результат побудови тесту для С17

Ітерація	Несправність	Входи (тест)					Несправності, що викриваються тестом	Наступна несправність
		1	2	3	6	7		
1	1^1	0	0	1	1	1	$1^1, 3^0, 11^1, 19_{11}^1, 16^0, 22^1, 23^1$	2^1
2	2^1	0	0	0	0	0	$2^1, 7^1, 16^0, 22^1, 23^1$	10_3^1
3	10_3^1	1	0	0	1	0	$2^1, 3^1, 10_3^1, 7^1, 16^0, 22^1, 23^1$	11_3^1
4	11_3^1	0	1	0	1	0	$3^1, 11_3^1, 11^0, 16^1, 22_{16}^1, 23_{16}^1, 22^0, 23^0$	6^1
5	6^1	0	1	1	0	0	$6^1, 11^0, 16^1, 22_{16}^1, 23_{16}^1, 22^0, 23^0$	10^1
6	10^1	1	0	1	0	1	$3^0, 6^1, 10^1, 11^0, 19^1, 22^0, 23^0$	16^1_{11}
7	16^1_{11}	0	1	1	1	1	$1^1, 3^0, 11^1, 16_{11}^1, 19_{11}^1, 16^0, 22^1, 23^1$	

Висновки. Запропонована методика може бути рекомендована для побудови повних перевіряючих тестів цифрових комбінаційних схем сучасних електронних пристроїв. Скорочення перебору методом фокусованого пошуку дає можливість застосовувати його для схем з більшою складністю, ніж інші детерміновані методи. Реалізація дедуктивного методу моделювання цифрових схем з несправностями на основі теоретико-множинного підходу суттєво підвищує швидкість отримання реакції схеми на входні сигнали одночасно в усіх її станах.

Перспективними вважаються наступні напрямки продовження досліджень:

- застосування результатів побудови повного тесту для локалізації несправності, тобто вирішення діагностичної задачі;
- продовження теоретичних і практичних досліджень з метою включення в запропоновану методику можливості будувати тести для цифрових схем з пам'яттю.

ЛІТЕРАТУРА

1. Куліков В.М. Порівняльний аналіз методів побудови тестів для цифрових пристроїв / Куліков В.М. / "Information Technology and Security". – Vol. 1, №2. – 2012. – с.34 – 44.
2. Куліков В.М. Підхід до побудови тестів перевірки цифрових пристроїв на надвеликих інтегральних схемах / Куліков В.М. / Information Technology and Security IC33I НТУУ «КПІ». – Vol. 1, № 1. – 2012. – с. 83 – 92.
3. Куліков В.М. Метод моделирования цифровых схем с неисправностями. / Куліков В.М., Кравчук В.В. / "Information Technology and Security". – Vol. 3, № 1. – 2015. – с. 50 – 59.
4. Ефанов Д.В. Коды с суммированием с фиксированными значениями кратностей обнаруживаемых монотонных и асимметричных ошибок для систем технического диагностирования / Ефанов Д.В., Сапожников В.В., Сапожников Вл.В. / Автоматика и телемеханика, вып. 6. – 2019. с.121 – 141.
5. Хаханов В.И. Кубитные структуры данных вычислительных устройств / Хаханов В.И., Ваджеб Гариби, Литвинова Е.И., Шкиль А.С. / Электронное моделирование. – Т. 37, № 1. – 2015. – с.76 – 99.
6. Киселев В.В. Метод диагностики цифровых схем с программируемыми ПЛИС на этапе изготовления / Киселев В.В., Суворов Н.А. / Электротехника, информационные технологии, системы управления. Вестник ПНИПУ. – № 14. – 2015. – с.97 – 107.
7. Лапутенко А.В. Обработка экспериментальных данных при верификации компонентов физических систем: оценка качества тестовых последовательностей / Лапутенко А.В., Лопез Х.Е., Евтушенко Н.В. / Известия высших учебных заведений. Физика. – Т. 60, №11. – 2017. – с.146-151.
8. Ермилов В.А. Метод отбора существенных неисправностей для диагностики цифровых схем. Общие выражения для неисправностей, возможных при эксперименте / Ермилов В.А. / Автоматика и телемеханика. – № 1. – 1971. – с. 159 – 167.
9. Ермилов В.А. Об алгоритме построения для логических сетей с памятью входных различающих последовательностей относительно заданного множества неисправностей / Ермилов В.А. / Автоматика и телемеханика. – № 3 – 1981. – с. 133 – 139.
10. Карибский В.В., Пархоменко П.П., Согомонян Е.С. Основы технической диагностики / Под ред. П.П.Пархоменко. – М. :Энергия, 1976, – 463с.
11. Kulikov V.M. On experimental research of efficiency of tests construction for combinational circuits by the focused search method / Kulikov V.M, Mokhor V.V. / 2011 9th East-West Design & Test Symposium (EWDTS), IEEE, 247 – 250.
12. Сапожников В.В. Об отношениях между неисправностями в комбинационных логических схемах / Сапожников В.В., Сапожников Вл.В. / Автоматика и телемеханика. – №1. – 1978. – с.167 – 171.