

*Куліков В.М.*

## **ПІДХІД ДО ПОБУДОВИ ТЕСТІВ ПЕРЕВІРКИ ЦИФРОВИХ ПРИСТРОЇВ НА НАДВЕЛИКИХ ІНТЕГРАЛЬНИХ СХЕМАХ**

### **Анотація:**

*Розглядається проблема скорочення перебору при побудові повних тестів перевірки цифрових пристроїв, розроблених із використанням надвеликих інтегральних схем. Процес побудови тесту для заданої несправності приводиться до вигляду пошуку термінальної вершини в дереві призначення сигналів. Для скорочення перебору пропонується застосовувати метод сфокусованого пошуку, розроблений як загальнотеоретичний метод пошуку рішення в системах продукційного типу. Описано результати експериментів із комбінаційними схемами.*

### **Аннотация:**

*Рассматривается проблема сокращения перебора при построении полных проверяющих тестов для цифровых устройств, разработанных с использованием сверхбольших интегральных схем. Процесс построения теста для заданной неисправности приводится к виду поиска терминальной вершины в дереве назначения сигналов. Для сокращения перебора предлагается применять метод сфокусированного поиска, разработанный как общетеоретический метод поиска решения в системах продукционного типа. Описаны результаты экспериментов с комбинационными схемами.*

### **Abstract:**

*Considered the problem of reducing the running in constructing a complete test for digital devices designed with using the VLSI circuits. The process of constructing a test for a given fault is reduced to finding the terminal vertices in the signals assignment tree. To reduce the iteration is proposed the focused search method wich designed to apply as the general theoretical method for finding solutions in productional type systems. Describes the results of experiments with the combinational circuits.*

### **Вступ**

Постійна тенденція до мікромініатюризації та зменшення енергоспоживання в сучасних технологіях виготовлення засобів електронної обчислювальної техніки приводить до стрімкого зростання складності елементної бази цифрових пристроїв. Показовими в даному сенсі можна вважати дані, оприлюднені на конференції "9th IEEE East-West Design & Test Symposium" [1]: складність сучасного смартфона iPhone 5 перевищує складність бортового комп'ютера космічного корабля Space Shuttle. Під складністю тут і далі розуміється кількість логічних елементів, з яких будується пристрій.

Структуровані вентильні матриці сімейства Nextreme-2 фірми eASIC, які почали з'являтися на ринку, починаючи з 2007 р., створені за 45нм технологією і включають до 20 млн. традиційних x вентилів (2AND-NO) та 1,9 млн. елементів пам'яті (типу eCell).

Відповідно до закону Гордона Мура (одного з засновників компанії Intel) рівень інтеграції мікросхем подвоюється кожні 1,5-2 роки [2]. Зростаюча складність цифрових

пристроїв має своїм наслідком підвищення складності діагностичного забезпечення цих пристроїв, а саме процедур моделювання і тестування, необхідних для пошуку дефектів на всіх етапах життєвого циклу, від проектування і виготовлення до технічного обслуговування і ремонту під час експлуатації.

Основою будь-якої процедури діагностування цифрового пристрою є аналіз його реакції на спеціальні вхідні сигнали, які називаються тестовими сигналами або просто тестами. Задача знаходження (побудови) тестів належить до класу NP-складних і її обчислювальна складність має експоненціальну залежність від складності пристрою [3]. Таке становище пояснюється значною кількістю як елементів, так і можливих дефектів. Якщо складність пристрою оцінити деяким числом  $N$  (як правило це кількість входів і виходів всіх логічних елементів), то кількість дефектів, для виявлення яких необхідно будувати тести, оцінюється числом  $3^N$ .

Усі методи побудови тестів, не зважаючи на різну математичну форму представлення, мають у своїй основі перебирання множин сигналів, які здатні забезпечити активізацію несправності (дефекту) на виході несправного елемента та транспортування викривленого значення до одного з виходів схеми [4]. При кількості входів схеми  $M$ , кількість комбінацій сигналів, які треба перебирати, оцінюється числом  $2^M$ . Причому ситуація, коли для одного дефекту потрібно знайти один набір тестових сигналів, існує тільки для комбінаційних схем. Для схем з пам'яттю, як правило, необхідно ще забезпечити переведення схеми в певний стан, в якому несправність проявляє себе на виходах схеми.

Згідно з даними, наведеними у [5], вартість верифікації програмно-апаратних продуктів на основі надвеликих інтегральних схем (ASIC, IPcore, SoC) складає 70% від загальних витрат на проектування. Розмірність testbench-коду може сягати 80% від загальної довжини формального опису проекту мовою HDL.

Складність задачі створення якісного діагностичного забезпечення для апаратури на мікросхемах із великим ступенем інтеграції логічних елементів спричиняє необхідність приділяти цьому питанню значну увагу ще на етапі проектування цифрових пристроїв. Зокрема, створенню нових технологій проектування – DFT (design for testability, Boundary-Scan, JTAG), призначених компенсувати обмеження керованості (controllability) та можливості спостереження (observability), необхідні при тестовій перевірці пристроїв. Основні вимоги технології граничного (Boundary-Scan, JTAG) сканування закріплені в стандартах IEEE Std 1149.1 та IEEE Std 1500.

Застосування субмікронних та нових молекулярних нанотехнологій при виготовленні мікросхем збільшує вірогідність виникнення сталих та несталих несправностей протягом усього життєвого циклу цифрового пристрою і тому потребує періодичного тестування. У реальних умовах застосування комплексів інформаційно-телекомунікаційних систем необхідна швидка перевірка окремих блоків апаратури із мінімальним порушенням функціонування всього комплексу в цілому [6].

Існуючі на сьогодні системи контролю будуються на таких підходах [7]:

- функціональний контроль в режимі on-line, при якому пристрій досліджується під час робочого функціонування на його робочі частоті;

- тестове діагностування в режимі off-line із застосуванням розроблених наборів тестових сигналів у заданому класі несправностей (константного обриву, замикання та ін.)

під час призупинення експлуатації об'єкту із використанням зовнішнього (Automatic Test Equipment - ATE) або вбудованого (Built-In Self-Test - BIST) діагностичного обладнання.

Останнім часом із розвитком методів проектування, орієнтованих на підвищення придатності пристроїв до тестування, спостерігається значне скорочення області застосування зовнішнього діагностичного обладнання для тестування систем. Перші позиції за обсягом застосування займають засоби вбудованого тестування. Такий підхід до забезпечення контролю не вимагає зовнішнього обладнання, яке в даному випадку інтегрується до складу інфраструктури кристалу. Зазвичай вбудовані засоби діагностування використовують методи компактного тестування, зокрема сигнатурного аналізу, заснованого на застосуванні регістрів зсуву з лінійним зворотним зв'язком. Тестові сигнали при цьому ніде не зберігаються, а генеруються автоматично.

Застосування методів вбудованого компактного тестування може бути корисним на всіх етапах життєвого циклу пристрою. Але істотним їх недоліком є те, що не гарантується повнота перевірки об'єкта діагностування. Тому одночасно із вбудованим компактним самотестуванням доцільно використовувати заздалегідь розроблену тестову послідовність, яка доповнює послідовність, генеровану автоматично.

Таким чином, проблема побудови тестів залишається актуальною одночасно із застосуванням різних методів забезпечення контролепридатності та вбудованого контролю об'єктів.

#### **Алгоритм побудови вхідної послідовності сигналів для перевірки несправностей у цифровій схемі**

Задачу побудови тестової послідовності вхідних сигналів цифрової схеми можна представити, як рішення двох задач:

задачі побудови вхідної послідовності, яка розрізняє пару технічних станів  $S_i, S_j$  цифрової схеми;

задачі пошуку стану  $S_k$  цифрової схеми, який не відрізняється реакцією схеми від справного стану на вже побудованій частині тестової послідовності сигналів.

Зазначені задачі відомі в технічній діагностиці як, відповідно, зворотна та пряма і вважаються класичними [8].

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 [9].

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

*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  $X$ :  $X = X \parallel T_k$ .

*Крок 3.* Знайти стан  $S_k$ , який не відрізняється від  $S_n$  реакцією на  $X$ . Якщо  $S_k$  існує, то перейти до кроку 2, інакше перейти до кроку 4.

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

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

У роботах [10,11] 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.

### Задача призначення сигналів цифрової схеми

Відомо, що всі методи побудови тестів цифрових схем мають у своїй основі некерований перебір у тій чи іншій формі, яка залежить від застосованого математичного апарата. З метою скорочення перебору методом сфокусованого пошуку задача побудови тесту для пари технічних станів цифрової схеми приводиться у форму задачі пошуку набору вхідних сигналів схеми, які породжують задані значення сигналів на виходах цієї схеми. У запропонованій у [12] формі ця задача має назву "задача призначення сигналів". Далі наведено стислий опис рішення цієї задачі для комбінаційної схеми.

Вважати, що:

елементи цифрової схеми пронумеровані цілими числами  $1, 2, \dots, n$ ;

$X = \{x_1, x_2, \dots, x_m\}$  – множина змінних, які представляють вхідні сигнали схеми;

$P = \{p_1, p_2, \dots, p_n\}$  – множина змінних, якими позначені виходи одновихідних елементів схеми.

Входам елементів приписані змінні виходів зв'язаних з ними елементів або вхідні змінні схеми.

Елементи схеми реалізують логічні функції (функції перемикавання) від змінних, якими позначені їх входи. Функція елемента  $i$  представляється у прямій та зворотній диз'юнктивній нормальній формі:

$$\Phi_i = \sum_j A_{ij}, \quad \overline{\Phi_i} = \sum_k A_{ik},$$

де  $A_{ij}, A_{ik}$  – кон'юнкції літер вхідних змінних елемента:

$$A_{ij}, A_{ik} = \prod_l v_{k,l}^{s(k)},$$

де  $v_{k,l}^{s(k)}$  – літера змінної  $v_k \in P \cup X$ , яка розташована на  $l$ -ом місці в кон'юнкціях  $A_{ij},$

$$A_{ik}, s(k)=0, 1, v_{k,l}^0 = \overline{v_{k,l}}, v_{k,l}^1 = v_{k,l}.$$

Алгоритм пошуку набору значень вхідних сигналів, на яких обрані сигнали мають задані значення, представляється у вигляді пошуку термінальної бази в системі продукційного типу, і має форму пошуку в дереві.

Вершиною дерева є кон'юнкція літер внутрішніх і вхідних змінних, які разом із кон'юнкціями вершин, що розташовані на шляху від кореневої вершини в дану вершину, забезпечують необхідний набір значень сигналів початкової (кореневої) вершини. Літери внутрішніх змінних – це літери, для яких необхідно зробити підстановки (призначення) для утворення наступних вершин. Кореневою вершиною дерева є кон'юнкція, яка визначає початковий набір значень сигналів, які необхідно забезпечити призначенням вхідних сигналів. Якщо набір значень сигналів вершини описується кон'юнкцією літер  $B$ , першою внутрішньою змінною якої є  $p_i$ , то з цієї вершини можна побудувати стільки наступних вершин, скільки кон'юнкцій в функції  $\Phi_i^{s(i)}$ . Тут  $s(i)=0, 1, \Phi_i^0 = \overline{\Phi_i}, \Phi_i^1 = \Phi_i$ . Кожній новій вершині відповідає кон'юнкція  $B'$ , яку отримано з кон'юнкції  $B$  попередньої

видаленням з неї  $p_i^{s(i)}$  та додаванням тих літер  $A_{i,j}, A_{i,k}$ , яких немає в  $B$  та у раніше створених вершинах. Термінальною є вершина, кон'юнкція літер якої не містить літер внутрішніх змінних схеми. Вершина є тупиковою, якщо у її кон'юнкції міститься літера, інверсна літері будь-якої з вершин, що розташовані на шляху від кореневої вершини в дану.

Приведення задачі побудови тесту для пари технічних станів  $S_n, S_j$ , де  $S_n$  – справний стан, до задачі призначення сигналів полягає:

- у формуванні за описом моделі справної схеми моделі схеми із заданою несправністю шляхом зміни функцій несправних у стані  $S_j$  елементів і присвоєння  $k$ -му елементу номера  $k+n$ , якщо сигнали цього елемента можуть відрізнитися в станах  $S_n$  та  $S_j$ ;

- у виконанні призначення сигналів з кореневої вершини, кон'юнкція якої складається з літер  $p_k^{s(k)}$  і  $\overline{p_{k+n}^{s(k+n)}}$ , для всіх  $p_k$  з множини вихідних змінних та  $s(k)$  з множини  $\{1,0\}$ , доки не буде знайдено вершину, що утворена тільки з вхідних змінних.

**Приклад.** Побудувати тест несправності  $b_0$  (закріплення нульового значення сигналу на виході 6-го елемента) в схемі контрприкладу Шнейдера [13], яку зображено на рис. 1.

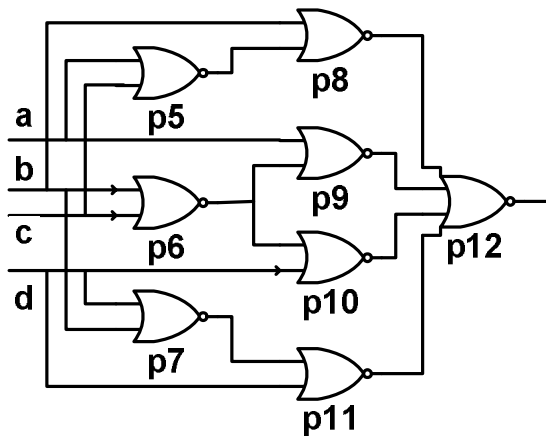


Рис. 1. Схема Шнейдера

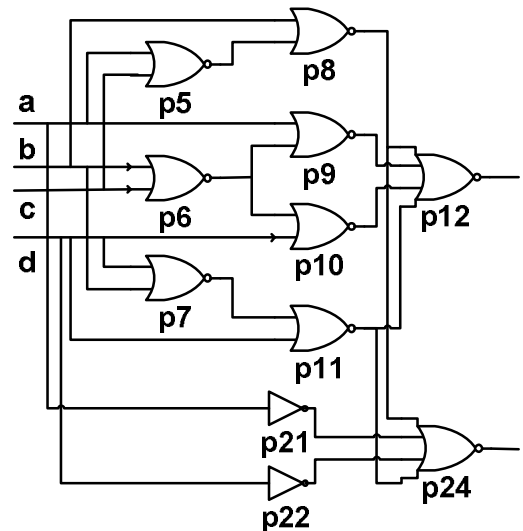


Рис. 2. Еквівалентна схема

Для побудови тесту вихідна схема перетворюється в еквівалентну схему, яка формується, виходячи з умов розпізнавання несправності на виходах схеми. Змінні, призначені виходам елементів справної схеми, утворюють множину  $\{p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}\}$ , змінні схеми з несправністю  $b_0$  утворюють множину  $\{p_5, p_7, p_8, p_{11}, p_{21}, p_{22}, p_{24}\}$ , вхідні змінні схеми в двох станах утворюють множину  $X = \{a, b, c, d\}$ . Справна схема і схема з несправністю  $b_0$  у формі еквівалентної схеми показані на рис. 2. Частина елементів не чутливі до впливу несправності; ці елементи утворюють спільну для двох схем частину. До неї входять елементи множини  $\{p_5, p_7, p_8, p_{11}, a, b, c, d\}$ . Тестом несправності  $b_0$  буде вважатися комбінація вхідних сигналів, яка забезпечить встановлення логічно протилежних значень сигналів на виходах елементів 12 та 24 еквівалентної схеми.

Дерево значень сигналів подано на рис. 3. Кореневою вершиною дерева є кон'юнкція  $p_{12} \overline{p_{24}}$ , яка визначає умови викриття несправності за вихідними сигналами.

Термінальна вершина 50 містить тест несправності  $\mathbf{6}_0 \{a, b, c, d\}$ .

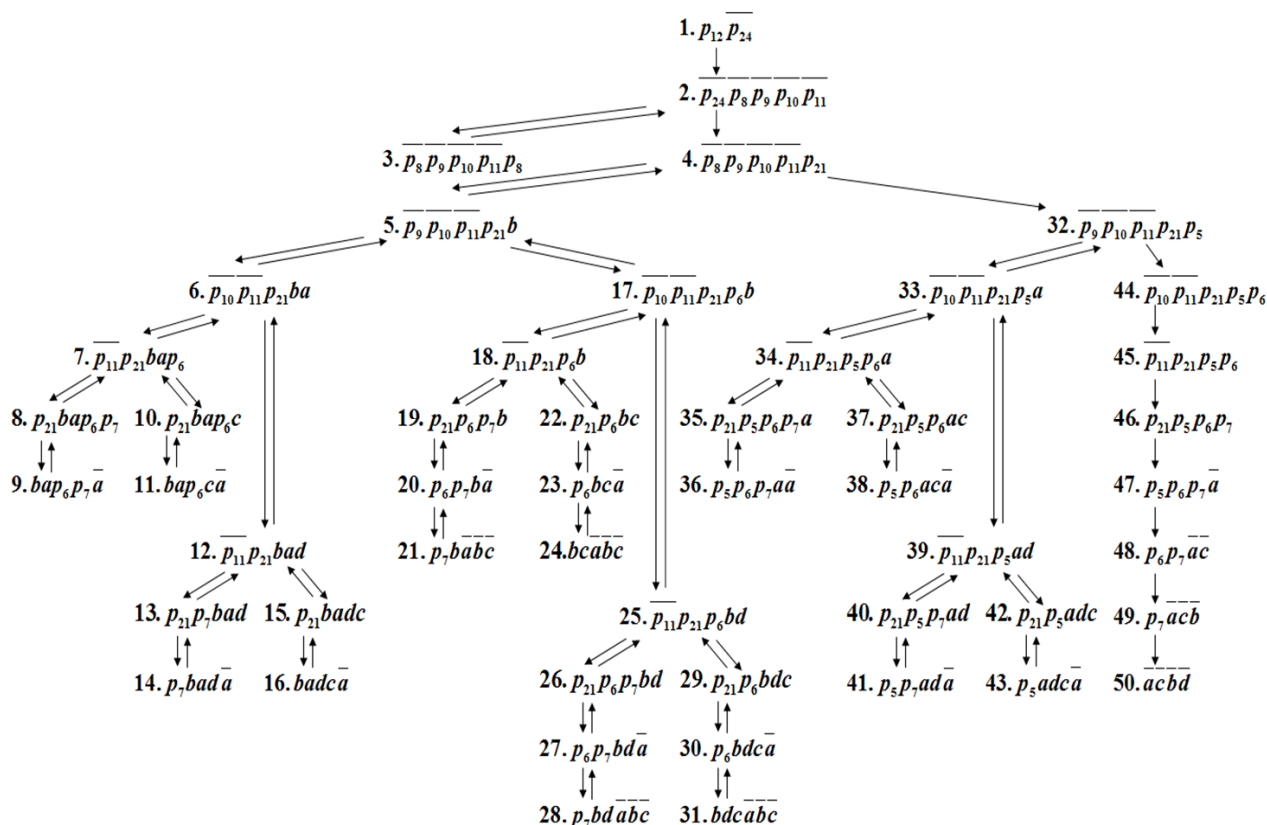


Рис. 3. Дерево значень сигналів без скорочення перебору

### Фокусований пошук рішення задачі призначення сигналів

Нехай  $q$  – літера з кон'юнкції  $B_{k(i)}$  вершини  $k_i$ ,  $A/q$  – підстановка кон'юнкції  $A$  замість літери  $q$ , яка породжує з вершини  $k_i$  тупикову вершину  $k_{i+1}$ .

*Часткова заборона* – це кон'юнкція літер раніше призначених сигналів, які спричинили виникнення протиріччя при підстановці  $A/q$ . Часткова заборона приписується дузі з вершини  $k_i$  у вершину  $k_{i+1}$ .

Вершина  $k_{i+1}$  є тупиковою внаслідок того, що кон'юнкція  $A$  містить літери  $\overline{v_1 v_2 \dots v_r}$ , які інверсні по відношенню до літер  $v_1 v_2 \dots v_r$  раніше призначених сигналів. Формально ці умови, які спричинили утворення тупикової вершини  $k_{i+1}$  представляються диз'юнкцією  $qv_1 + qv_2 + \dots + qv_r$ . Цей вираз представляє часткову заборону  $Z_{k(i), k(i+1)}$  для дуги в тупикову вершину  $k_{i+1}$ .

Нехай тепер все  $t$  вершин, які породжуються з вершини  $k_i$ , є тупиковими. Кон'юнкції літер, які роблять тупиковими кожен з вершин  $k_{i+1}, k_{i+2}, \dots, k_{i+t}$  окремо, містяться в часткових заборонах  $Z_{k(i), k(i+1)}, Z_{k(i), k(i+2)}, \dots, Z_{k(i), k(i+t)}$  відповідно. Тоді кон'юнкції літер, які роблять всі вершини  $k_{i+1}, k_{i+2}, \dots, k_{i+t}$  тупиковими одночасно, представляються виразом

$$Z_{k(i)} = Z_{k(i), k(i+1)} \& Z_{k(i), k(i+2)} \& \dots \& Z_{k(i), k(i+t)},$$

приведеним до диз'юнкції кон'юнкцій. Кожна з кон'юнкцій  $Z_j$  виразу для  $Z_{k(i)}$  є заборонаю. Заборони  $Z_j$ , які формуються під час пошуку термінальної вершини, накопичуються у вигляді диз'юнкції кон'юнкцій виразу  $Z$ , значення якого спочатку встановлюється рівним нулю.

Вершина  $k_i$  задовольняє умовам заборони  $Z_j$  і, таким чином, є тупиковою, якщо множина літер вершини  $k_i$  і літер вершин на шляху з кореневої вершини в дану містить усі літери кон'юнкції  $Z_j$ . Якщо  $A/q$  – підстановка, результатом якої є утворення тупикової вершини  $k_i$  з вершини  $k_{i-1}$ ,  $B_{k(i-1)}$  – кон'юнкція літер вершини  $k_{i-1}$ , то часткова заборона для дуги  $k_{i-1}k_i$  визначається як  $Z_{k(i-1),k(i)} = q(B_{k(i-1)} Z_j)$ , де  $B_{k(i-1)} Z_j$  – кон'юнкція літер, спільних для  $Z_j$  і всіх кон'юнкцій вершин від кореневої до вершини  $k_{i-1}$  включно. Часткова заборона  $Z_{k(i-1),k(i)}$  утворюється з літер кон'юнкції заборони  $Z_j$  включенням до неї літери  $q$  і видаленням літер кон'юнкції  $A$ , які у вершині  $k_i$  призначалися вперше.

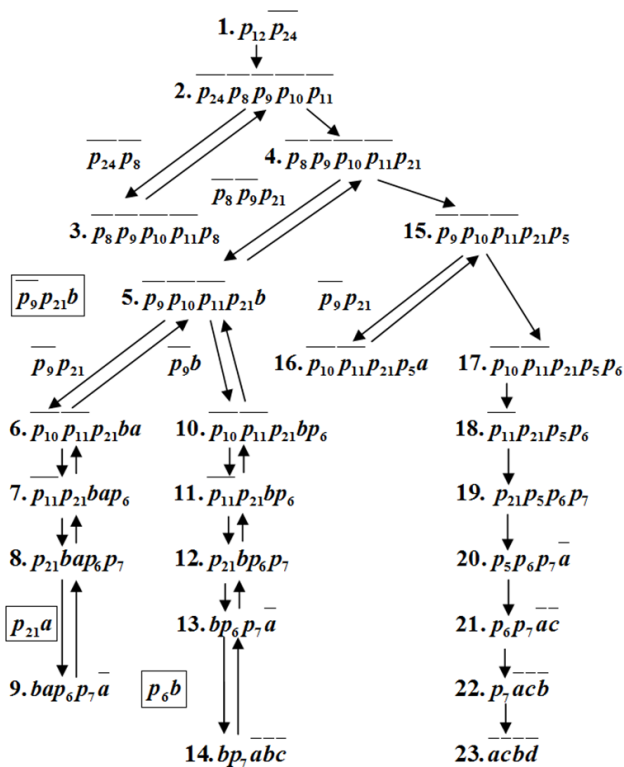


Рис. 4. Дерево скороченого перебору

Для дуги  $k_j k_{j+1}$  формується часткова заборона  $Z_{k(j),k(j+1)}$  і виконується спроба побудувати чергову вершину ярусу  $j+1$ .

Приклад. Використовуючи описаний метод, знайдемо набір значень вхідних сигналів, на яких перевіряється несправність  $6_0$  в схемі контрприкладу Шнейдера. Для цього скористуємось схемою, зображеною на рис. 2. Дерево призначених сигналів подано на рис. 4. Стосовно дерева повного некерowanego перебору, яке складалося з 50 вершин, застосування методу сфокусованого пошуку дозволило скоротити кількість вершин до 23.

### Результати експериментів з комбінаційними схемами

Описаний метод засновано на представленні процесу побудови тестів в формі пошуку рішення задачі призначення сигналів. При цьому, можна довести, що "Information Technology and Security" № 1(1)-2012

обчислювальна складність пошуку рішення задачі призначення сигналів дорівнює складності всіх інших детермінованих методів побудови тестів. Тому для оцінки ефективності методу сфокусованого пошуку відносно інших методів достатньо виконати порівняння величини перебору в задачі призначення сигналів при сфокусованому пошуку рішення і при некерованому переборі.

Для оцінки міри скорочення перебору при сфокусованому пошуку порівняно з некерованим перебором проведено ряд експериментів. Для цього було обрано схему 32-розрядного суматора з прискореним переносом. Схема складається з восьми 4-розрядних комбінаційних суматорів і трьох схем прискореного переносу. Вибір обумовлено, по-

перше, загальною відомістю та доступністю схеми, по-друге, можливістю нарощувати складність до отримання схеми будь-якої складності. Складність 32-розрядного суматора складає 659 логічних елементів.

В якості міри величини перебору обрано  $N$  – кількість вершин в дереві призначення сигналів. Побудова тестів для кожної випадково обраної несправності виконувалась двома програмами, перша з яких виконує призначення сигналів із використанням методу сфокусованого пошуку, а друга – побудову повного дерева некерованого перебору. Кількість побудованих до знаходження рішення вершин для першої і другої програм показано на графіку рис. 5 у логарифмічному масштабі.

Вісь  $X$  показує кількість вершин у дереві скороченого перебору, а вісь  $Y$  – кількість вершин, які побудовано як першою так і другою програмою. Для кожної несправності на графіку зображено дві мітки з

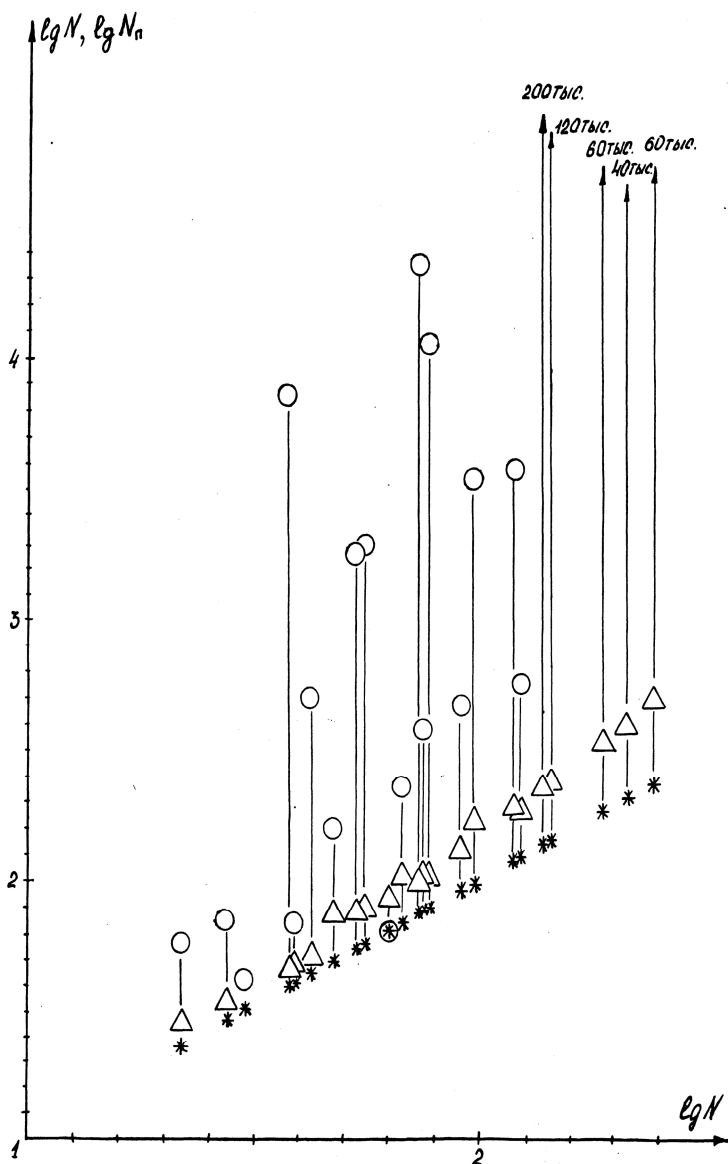


Рис. 5. Результати експериментів

однією координатою  $X$  і різними координатами  $Y$ . Відміткою  $*$  показано кількість вершин у скороченому дереві, знаком  $O$  – кількість вершин в дереві некерованого перебору. Абсолютна величини виграшу у кількості вершин від застосування методу



сфокусованого пошуку дорівнює різниці між значеннями координат  $Y$  двох відміток для однієї несправності. На графіку можна бачити тенденцію до збільшення виграшу із збільшенням розміру дерева некерованого перебору.

Для оцінки витрат на управління пошуком на графіку знаком  $\Delta$  показано кількість вершин, яку побудувала б програма некерованого перебору за час роботи програми із фокусуванням пошуку. Можна бачити, що витрати на реалізацію сфокусованого пошуку незначні порівняно із виграшем у кількості вершин.

Отримані результати дозволяють зробити наступні висновки.

### Висновки

1. Тенденція до збільшення складності цифрових пристроїв при використанні мікросхем із надвеликим ступенем інтеграції ускладнює задачу розробки їх діагностичного забезпечення. Це пов'язано із обмеженням доступу до внутрішньої інфраструктури мікросхем і розвитком нових технологій проектування, які мають забезпечити урахування вимог придатності цифрового пристрою для тестового контролю на всіх етапах життєвого циклу.

2. Нові технології проектування цифрових пристроїв орієнтовані передусім на використання засобів вбудованого контролю і діагностування. Як правило вбудовані засоби використовують або генератори псевдовипадкових сигналів-тестів, або вичерпне тестування, які не забезпечують достатньої повноти перевірки або не можуть вважатися ефективними. Крім того, виникає проблема контролю самого вбудованого діагностичного обладнання, яке також може бути несправним. Тому більш ефективним може вважатися використання комплексних систем вбудованого псевдовипадкового тестування із тестами, побудованими детермінованими методами.

3. Усі відомі детерміновані методи побудови тестів мають у своїй основі повний перебір усіх множин вхідних сигналів до знаходження комбінації, яка забезпечує активізацію і транспортування несправності до одного з виходів схеми. Тому практичний інтерес можуть представляти дослідження, спрямовані на зменшення перебору в детермінованих методах генерації тестів, зокрема застосування для цього методу скорочення перебору, який отримав назву "метод сфокусованого пошуку". Скорочення перебору при побудові тестів, дасть можливість підвищити поріг складності цифрових схем, для яких можна побудувати набори вхідних сигналів, що перевіряють 100% несправностей.

4. У роботі розглядається лише рішення задачі побудови тесту для пари несправних станів цифрової комбінаційної схеми (зворотна задача). Пряму задачу, як правило, відносять до задач моделювання поведінки цифрової схеми з несправностями.

### Література:

1. *Rich Goldman*. Tech & Space: A Symbiotic Relationship // 9th IEEE East-West Design & Test Symposium. Final program. – 2011. – P.15.
2. *Угрюмов Е.П.* Цифровая схемотехника: учебн. пособие для вузов. – 3-е изд., – СПб.: БХВ-Петербург. – 2010. – 816 с.: ил.
3. *Ibarra O.H., Sahni S.* Polynomially complete fault detection problems // IEEE Trans. Comput. – 1975. – V. C-24. 3. – P. 242-249.

4. *Roth J.P.* Diagnosis of automata failures: a calculus and a method // *IEEE Trans. Comput.* – 1966. – V. 15. – № 7. – P. 278-291.
5. *В.И. Хаханов, И. В. Хаханова, Е.И. Литвинова, О.А. Гузь.* Верификация HDL-кода на основе механизма ассерций // *Інформаційно-керуючі системи на залізничному транспорті.* – 2009. – № 6. – С. 38-51.
6. *Г.П. Аксёнова.* Контролепригодная архитектура для самотестирования в программируемых логических матричных структурах // *Автоматика и телемеханика.* – 2010. – № 10. – С. 154-164.
7. *Л.А. Золоторевич, А.В. Ильинкова.* Разработка тестов для анализа контролепригодности СБИС на верхних уровнях проектирования // *Автоматика и телемеханика.* – 2010. – № 9. – С. 162-173.
8. *Карибский В.В., Пархоменко П.П., Согомоян Е.С.* Основы технической диагностики / Под ред. П.П. Пархоменко. – М.: Энергия, 1976. – 463 с.
9. *Ермилов В.А.* Об алгоритме построения для логических сетей с памятью входных различающих последовательностей относительно заданного множества неисправностей. // *Автоматика и телемеханика.* – 1981. – № 3. – С. 133-139.
10. *Сапожников В.В., Сапожников Вл.В.* Об отношениях между неисправностями в комбинационных логических схемах // *Автоматика и телемеханика.* – 1978. – № 1. – С. 167-171.
11. *Hughes Joseph L.A., Mc Clusky E.J.* Multiple stuck-at coverage of single stuck-at fault test sets. // *International Test Conference Proc., Washington, D.C.* – Sept.8-11. – 1986. – P. 368-374.
12. *Ермилов В.А.* Фокусированный поиск решения задач технической диагностики цифровых схем // *Автоматика и телемеханика.* – 1989. – № 8. – С. 144-154.
13. *Schneider P.R.* On the Necessity to Examine D-Chains in Diagnostic Test Generation - An Example. - *IBM Journal of Research and Development.* – 1967. – № 11(1). – P. 114.