УДК 681.326.7

Куліков В.М.

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ПОБУДОВИ ТЕСТІВ ДЛЯ ЦИФРОВИХ ПРИСТРОЇВ

Анотація:

Розглядається проблема низької ефективності відомих детермінованих методів побудови тестів для цифрових схем. Показано, що головною причиною їх непридатності для складних схем є некерований перебір всіх комбінацій сигналів, які забезпечують прояв несправності на виході елемента з дефектом, та сигналів, які забезпечують транспортування викривленого сигналу до одного з виходів схеми. Результатом поведеного аналізу є висновок стосовно надання переваги при побудові тестів методу фокусованого пошуку.

Анотация:

Рассматривается проблема низкой эффективности известных детерминированных методов построения тестов для цифровых схем. Показано, что главной причиной их непригодности для сложных схем является неуправляемый перебор всех комбинаций сигналов, которые обеспечивают проявление неисправности на выходе элемента с дефектом, и сигналов, которые обеспечивают транспортировку искаженного сигнала к одному из выходов схемы. Результатом поведенного анализа является вывод о предпочтительности выбора метода фокусируемого поиска для построения тестов.

Annotation:

Considers the problem of low efficiency of the known deterministic methods for constructing tests for digital circuits. Showed that the main reason of their unsuitability for complex circuits is uncontrolled search of the all combinations of signals, that provide the display of fault on the output of faulty element, and the signals, that provide transporting of the distorted signal to one of circuits output. The result of the analysis is the conclusion about the preference of choice of the focused search method for tests construction.

Вступ

З науки "Технічна діагностика" відомо, що проблема побудови повного перевіряючого теста для цифрового пристрою, насамперед потребує вирішення задачі побудови набору вхідних сигналів, по реакції на яку можливо розрізнити пару технічних станів S_iS_j даного пристрою. На сьогоднішній день розроблено достатньо велику кількість методів її вирішення. Всі методи розроблені для комбінаційних схем. Перехід до схем з пам'яттю полягає у представленні схеми у вигляді ітеративної комбінаційної моделі [1,2]. Побудова тестів для такої моделі виконується тими ж методами, що і для комбінаційних схем. Тому оцінку складності методів побудови тестів для пари технічних станів цифрового пристрою буде виконано для схем, які не містять зворотних зв'язків.

Моделлю цифрового пристрою (схеми) є логічна мережа $C=(\underline{x},\underline{g},\underline{y})$ [3], де $\underline{x}=(x_1,x_2,...,x_m)$ - вектор, складений зі змінних $\boldsymbol{x_i}$, якими позначені вхідні полюси (входи) схеми;

$\underline{y}=(y_1, y_2,..., y_n)$ - вектор змінних y_j, якими позначені вихідні полюси (виходи) схеми; $\underline{g}=(g_1(v_1), g_2(v_2),..., g_s(v_s))$ - вектор-функція, складений з булевих функцій $w_k = g_k(v_k)$. Функція g_k описує залежність значення вихідної змінної w_k логічного елемента k від значень змінних $v_{k_1}, v_{k_2},..., v_{k_{m_k}}$, які представляють його входи.

Щоб знайти тест для перевірки в заданій схемі несправності φ_r, достатньо для двох екземплярів мережі – справної $(\underline{x}, \underline{g}, \underline{y})$ і з заданою несправністю $(\underline{x}, \underline{g^r}, \underline{y^r})$, обчислити симетричні від'ємності $d_1 = (y_1 \oplus y_1^r)$, $d_2 = (y_2 \oplus y_2^r)$, ... , $d_n = (y_n \oplus y_n^r)$.

Тестом для несправності φ_r буде такий вектор вхідних сигналів $\underline{x}$, на якому встановлюється в одиницю функція: $h(y) = d_1 \vee d_2 \vee ... \; d_n$.

З теорії вибору [4] відомо, що пошук рішення будь-якої задачі виконується на деякій множині допустимих рішень (траєкторій), серед яких лише частина задовольняє умовам пошуку. При цьому складність вирішення (обчислювальна складність) оцінюється максимальною кількістю траєкторій, яка може бути отримана в процесі пошуку.

В задачі побудови теста для пари технічних станів логічної схеми, яка має $\boldsymbol{m}$ входів, траєкторіями є набори значень вхідних сигналів. Загальна кількість наборів дорівнює $\boldsymbol{2^m}$. В статті буде доведено, що жоден з відомих методів не знижує вказану оцінку складності вирішення задачі побудови тестів.

Методи побудови тестів

По способу формування наборів вхідних тестових сигналів для пари технічних станів логічної мережі всі методи поділяються на дві групи: методи випадкового пошуку і детерміновані (спрямовані) методи.

При випадковому пошуку тести обираються з випадкових послідовностей вхідних сигналів. Покриваюча здатність теста визначається одним з відомих методів моделювання схем з несправностями [2]. Характерним для випадкового пошуку є те, що спочатку за нетривалий термін часу до складу таких, що викриваються, потрапляє велика кількість несправностей, в основному тих, для яких існує досить велика кількість тестових наборів. Далі ефективність пошуку швидко знижується оскільки залишаються несправності, які перевіряються невеликою кількістю тестів і тому вірогідність їх випадкової генерації досить мала. Це є причиною того, що для схем складної структури випадковий пошук не забезпечує необхідної повноти тестів. Для схем з пам'яттю задача ускладнюється тим, що окремі несправності можуть проявлятися тільки на вхідних наборах, які повторюються з певною періодичністю. Оскільки період і кількість повторів для кожної несправності невідомі, то задача побудови тестів випадковим методами для схем із зворотними зв'язками є алгоритмічно нерозв'язною [5].

На сьогоднішній день запропоновано і поширено використовується велика кількість різноманітних прийомів звуження області пошуку випадкових вхідних наборів. Проте висновки щодо їх ефективності зроблено на підставі експериментів зі схемами певних розмірів і структури. Вцілому ж, питання про доцільність застосування різних стратегій генерації випадкових вхідних наборів досліджено недостатньо.

Очевидно, що складність вирішення задачі побудови теста для пари технічних станів комбінаційної схеми методом випадкової генерації дорівнює 2^m, де m – кількість вхідних полюсів мережі.

При побудові тестів будь-яким з детермінованих методів вхідні набори не обираються випадковим чином, а обчислюються згідно певних правил, виходячи з умов розрізнюваності станів по реакції схеми хоча б на одному з її виходів. Серед детермінованих методів найбільш поширено застосовуються D-алгоритм Рота [6], метод еквівалентних нормальних форм [7], метод булевих різниць [8] і метод розрізняючої функції [9].

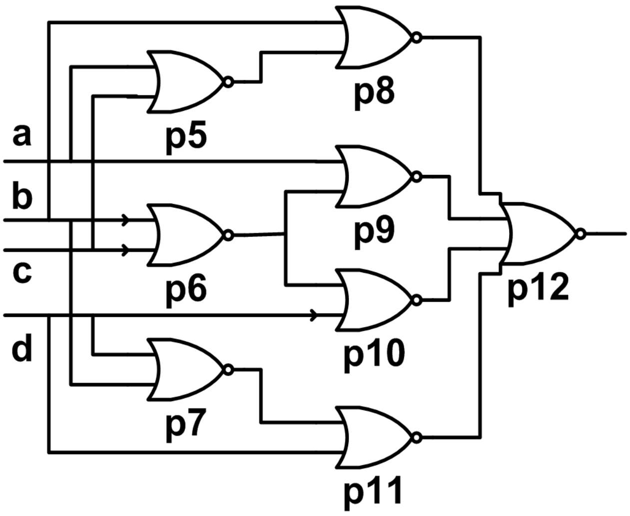

Рис. 1. Схема Шнейдера

Теорема. Обчислювальна складність D-алгоритма, метода булевих різниць, метода еквівалентних нормальних форм і метода розрізняючої функції дорівнює 2^m, де m – кількість вхідних полюсів комбінаційної схеми, що досліджується.

Справедливість твердження теореми буде доведено далі в цій статті. Текст статті ілюстровано прикладами побудови теста для виявлення несправності $\boldsymbol{6_0}$ (константний 0 на виході елемента 6) в схемі контрприклада Шнейдера [10] (Рис.1).

D-алгоритм

D-алгоритм являє собою метод визначення теста за допомогою знаходження шляху, по якому викривлення сигналу несправністю передається на вихід схеми. Такий шлях називається *активізованим*. Вибір шляху розповсюдження впливу несправності виконується за допомогою операції *D-перетину D-кубів* [6]. Для позначення операції введено символ $\bigcap_D$. Ідея використання D-кубів полягає у тому, щоб примусити один або декілька входів логічного елемента повністю визначати значення виходу цього елемента. Результатом D-перетину D-кубів послідовно з'єднаних елементів схеми є D-куб, який описує активізований шлях, що проходить через ці елементи.

1. $tc = \dfrac{2\;3\;6}{0\;0\;D}$

2. $tc = tc \bigcap_D C_9^1(1) = \dfrac{1\;2\;3\;6\;9}{0\;0\;0\;D\;\overline{D}}$

3. $tc = tc \bigcap_D C_{10}^1(1) = \dfrac{1\;2\;3\;4\;6\;9\;10}{0\;0\;0\;0\;D\;\overline{D}\;\overline{D}}$

4. $tc = tc \bigcap_D C_{12}^1(1) = \psi$

5. $tc = tc \bigcap_D C_{12}^1(2) = \psi$

6. $tc = tc \bigcap_D C_{12}^1(3) = \psi$

7. $tc = tc \bigcap_D C_{12}^1(4) = \psi$

8. $tc = tc \bigcap_D C_{12}^2(1) = \psi$

9. $tc = tc \bigcap_D C_{12}^2(1) = \dfrac{1\;\;2\;\;3\;\;4\;\;6\;\;8\;\;9\;\;10\;\;11\;\;12}{0\;\;0\;\;0\;\;0\;\;D\;\;0\;\;\overline{D}\;\;\overline{D}\;\;0\;\;D}$

10. $tc = tc \bigcap_D G_{11}(1) = \dfrac{1\;\;2\;\;3\;\;4\;\;6\;\;7\;\;8\;\;9\;\;10\;\;11\;\;12}{0\;\;0\;\;0\;\;0\;\;D\;\;1\;\;0\;\;\overline{D}\;\;\overline{D}\;\;0\;\;D}$

11. $tc = tc \bigcap_D G_8(1) = \dfrac{1\;\;2\;\;3\;\;4\;\;5\;\;6\;\;7\;\;8\;\;9\;\;10\;\;11\;\;12}{0\;\;0\;\;0\;\;0\;\;1\;\;D\;\;1\;\;0\;\;\overline{D}\;\;\overline{D}\;\;0\;\;D}$

12. $tc = tc \bigcap_D G_7(3) = \dfrac{1\;\;2\;\;3\;\;4\;\;5\;\;6\;\;7\;\;8\;\;9\;\;10\;\;11\;\;12}{0\;\;0\;\;0\;\;0\;\;1\;\;D\;\;1\;\;0\;\;\overline{D}\;\;\overline{D}\;\;0\;\;D}$

13. $tc = tc \bigcap_D G_5(3) = \dfrac{1\;\;2\;\;3\;\;4\;\;5\;\;6\;\;7\;\;8\;\;9\;\;10\;\;11\;\;12}{0\;\;0\;\;0\;\;0\;\;1\;\;D\;\;1\;\;0\;\;\overline{D}\;\;\overline{D}\;\;0\;\;D}$

Рис. 2. D-алгоритм

Для забезпечення розповсюдження впливу несправності через активізований шлях до одного з виходів схеми, необхідно забезпечити існування в схемі сигналів, які мають у відповідних кубах значення 0 або 1. З цією метою вводиться поняття *виродженого покриття* [6]. За допомогою операції D-перетину вироджених покриттів визначаються вхідні сигнали, які забезпечують існування активізованого шляху.

Таким чином, D-алгоритм побудови тестів зводиться до виконання операції D-перетину тестового куба ***tc*** з кубами елементів схеми у наступній послідовності.

Таблиця 1.

D-куби

	1	2	3	4	5	6	7	8	9	10	11	12
C_5^1	0		D		$\overline{D}$							
	D		0		$\overline{D}$							
C_6^1		0	D			$\overline{D}$						
		D	0			$\overline{D}$						
C_7^1		0		D			$\overline{D}$					
		D		0			$\overline{D}$					
C_8^1		0			D			$\overline{D}$				
		D			0			$\overline{D}$				
C_9^1	0					D			$\overline{D}$			
	D					0			$\overline{D}$			
C_{10}^1				0		D				$\overline{D}$		
				D		0				$\overline{D}$		
C_{11}^1			0				D				$\overline{D}$	
			D				0				$\overline{D}$	
C_{12}^1								D	0	0	0	$\overline{D}$
								0	D	0	0	$\overline{D}$
								0	0	D	0	$\overline{D}$
								0	0	0	D	$\overline{D}$
C_{12}^2								D	D	0	0	$\overline{D}$
								0	D	D	0	$\overline{D}$

1. Тестовий куб ***tc*** заповнюється значеннями сигналів D-кубу несправності, для якої будується тест.

2. Виконується побудова активізованого шляху від несправного елемента до будь-якого виходу схеми. Приєднання чергового елемента до активізованого шляху виконується за

допомогою D-перетину ***tc*** із D-кубом даного елемента. Ця фаза алгоритму називається *D-проходом*.

3. Визначаються сигнали, які забезпечують активізацію побудованого шляху. Цей процес називається *довизначенням* і виконується за допомогою D-перетину тестового куба ***tc*** із кубами виродженого покриття елементів. Якщо операція довизначення сигналів неможлива внаслідок виникаючих протиріч, то знайдений шлях бракується. В даному випадку виконується побудова і довизначення наступного шляху. При успішному завершенні операції довизначення тест для заданої несправності складається з вхідних сигналів схеми, які мають в ***tc*** значення 0 або 1.

Таблиця 2.

Куби виродженого покриття

	1	2	3	4	5	6	7	8	9	10	11	12
G_5	x		1		0							
	1		x		0							
	0		0		1							
G_6		x	1			0						
		1	x			0						
		0	0			1						
G_7		x		1			0					
		1		x			0					
		0		0			1					
G_8		x			1			0				
		1			x			0				
		0			0			1				
G_9	1					x			0			
	x					1			0			
	0					0			1			
G_{10}				1		x				0		
				x		1				0		
				0		0				1		
G_{11}			x				1				0	
			1				x				0	
			0				0				1	
G_{12}								x	x	x	1	0
								x	x	1	x	0
								x	1	x	x	0
								1	x	x	x	0
								0	0	0	0	1

Приклад. Процес побудови теста несправності ***$б_0$*** в схемі контрприклада Шнейдера показано на рис.2. З метою підкреслити перебірний характер алгоритму процес представлено в формі дерева. Послідовність операцій вказано стрілками. D-куби і куби виродженого покриття приведено відповідно у таблицях 1 і 2. Для D-кубів використовується позначення $C_i^k(l)$ і для кубів виродженого покриття - $G_i(l)$, де k – для одиничних кубів має значення 1, для подвійних – 2 і т.д., i – номер елемента, l – номер куба.

Лема 1. Обчислювальна складність D-алгоритму побудови теста для перевірки в комбінаційній схемі з m входами визначеної несправності дорівнює 2^m.

Доказ. У випадку, коли тест для визначеної несправності не існує, в D-алгоритмі буде необхідно перебирати всі шляхи від місця знаходження несправного елемента до кожного з виходів схеми. Не виключено, що всі шляхи активізації будуть відбраковані в фазі довизначення алгоритму при перетині ***tc*** з кубами найближчого до входів ярусу елементів схеми. В результаті активізації різних шляхів буде виконано перебирання всіх можливих для кожної конкретної схеми комбінацій кубів виродженого покриття елементів, входами яких є входи схеми. Куби виродженого покриття описують всі можливі набори значень вхідних сигналів елемента, а їх комбінації – все можливі набори значень сигналів схеми. Тому кількість різних вхідних наборів, яка може бути отримана при побудові теста із застосування D-алгоритму, оцінюється величиною 2^m. Твердження леми доведено.

Метод булевих різниць

Метод булевих різниць побудований на понятті булевої похідної (різниці) функції $f(x) = f(x_1, x_2, \ldots, x_j, \ldots, x_n)$, яка визначається виразом

$$df(x)/dx_j = f(x_1, x_2, \ldots, x_j, \ldots, x_n) \oplus f(x_1, x_2, \ldots, \bar{x}_j, \ldots, x_n)$$

Булева похідна встановлює значення змінних $x_1, x_2, \ldots, x_n$, за винятком x_j, при яких зміна значення x_j призводить до зміни значення функції $f(x)$. Тестом несправності $x_j \equiv 0$ ($x_j \equiv 1$) є значення логічних змінних, при яких $x_j \cdot df(x)/dx_j = 1$ ($\bar{x}_j \cdot df(x)/dx_j = 1$).

Якщо несправністю є інвертування значення внутрішнього сигналу схеми, то x_j розглядається у якості додаткової змінної і є умовою прояву несправності, вираженою через вхідні змінні. Так умовою прояву несправності 6_0 в схемі контрприкладу Шнейдера (рис.1) є вираз $x_6 = \bar{b}\,\bar{c}$. Умова означає, що несправність може бути виявлена, якщо вона інвертує значення сигналу на виході шостого елемента.

Приклад. Знайдемо методом булевих різниць вхідні набори для перевірки несправності $\boldsymbol{6_0}$ в схемі контрприклада Шнейдера. Вираз для функції $f(x)$ можна побудувати методом зворотної підстановки по структурі логічної мережі.

$$f(x) = x_6(bc + \bar{a}\bar{b}\bar{c}\bar{d}) + abcd$$

Похідна від $f(x)$ по x_6:

$$df(x)/dx_6 = \bar{a}bc + bc\bar{d} + \bar{a}\bar{b}\bar{c}\bar{d}$$

При обчисленні $df(x)/dx_6$ було застосовано властивості булевої похідної наведені в [1].

По булевій похідній функції $f(x)$ можна визначити тести несправності, яка аналізується:

$$x_6 \cdot df(x)/dx_6 = \bar{b}\bar{c} \cdot df(x)/dx_6 = \bar{a}\bar{b}\bar{c}\bar{d}$$

Таким чином несправність $\boldsymbol{6_0}$ перевіряється тестом ***0000***.

Лема 2. Складність вирішення задачі побудови теста для пари технічних станів комбінаційної логічної мережі з m входами методом булевих різниць дорівнює 2^m.

Доказ. Складність вирішення задачі можна визначити шляхом зведення даної задачі, до задачі, складність якої відома [4]. Побудова теста методом булевих різниць полягає у пошуку коренів булевого рівняння $x_j \cdot df(x)/dx_j = 1$. Відомо, що будь-який булевий вираз може бути представлений у кон'юнктивній нормальній формі (КНФ). Тому задача знаходження коренів рівняння $x_j \cdot df(x)/dx_j = 1$, в якому ліва частина представлена в КНФ, є ні чим іншим, як добре відомою *NP*-повною задачею про перевірку виконуваності КНФ [11]. Складність вирішення даної задачі оцінюється загальною кількістю можливих наборів значень двійкових змінних, що входять в КНФ. Якщо кількість змінних дорівнює m, то складність оцінюється величиною 2^m. Таким чином, якщо $f(x)$ та x_i виражені через m вхідних змінних, то задача побудови теста методом булевих різниць є *NP*-повною і складність її вирішення дорівнює 2^m. Твердження леми доведено.

Метод еквівалентних нормальних форм (ЕНФ)

Ідея методу ЕНФ полягає у пошуку шляхів транспортування впливу несправності на основі логічних виразів, які описують функціонування цифрової схеми. Пошук теста несправності починається з побудови ЕНФ функції, яку реалізує дана схема. Дана операція виконується по еквівалентному дереву (ЕД) або скобковій формі [1]. ЕД для схеми контрприклада Шнейдера представлене на рис.3.

ЕНФ Z і зворотна ЕНФ (ЗЕНФ) $\overline{Z}$ будуються по ЕД за допомогою процедури зворотної підстановки. Побудовані по ЕД на рис.3 ЕНФ і ЗЕНФ мають вигляд:

$$
\begin{aligned}
Z = {} & a_2 b_1 \overline{b_3}\,\overline{b_4}\,\overline{c_3}\,\overline{d_2} + a_2 b_1 \overline{b_3}\,\overline{c_3} c_4 + a_2 b_1 \overline{b_4} d_1 \overline{d_2} + a_2 b_1 c_4 d_1 + b_1 \overline{b_2}\,\overline{b_3}\,\overline{b_4}\,\overline{c_2}\,\overline{c_3}\,\overline{d_2} + b_1 \overline{b_2}\,\overline{b_3}\,\overline{c_2} c_3 c_4 + \\
& + b_1 \overline{b_2}\,\overline{b_4}\,\overline{c_2} d_1 \overline{d_2} + b_1 \overline{b_2}\,\overline{c_2} c_4 d_1 + \overline{a_1} a_2 \overline{b_3}\,\overline{b_4}\,\overline{c_1}\,\overline{c_3}\,\overline{d_2} + \overline{a_1} a_2 \overline{b_3}\,\overline{c_1}\,\overline{c_3} c_4 + \overline{a_1} a_2 \overline{b_4}\,\overline{c_1} d_1 \overline{d_2} + \overline{a_1} a_2 \overline{c_1} c_4 d_1 + \\
& + \overline{a_1}\,\overline{b_2}\,\overline{b_3}\,\overline{b_4}\,\overline{c_1}\,\overline{c_2}\,\overline{c_3}\,\overline{d_2} + \overline{a_1}\,\overline{b_2}\,\overline{b_3}\,\overline{c_1}\,\overline{c_2}\,\overline{c_3} c_4 + \overline{a_1}\,\overline{b_2}\,\overline{b_4}\,\overline{c_1}\,\overline{c_2} d_1 \overline{d_2} + \overline{a_1}\,\overline{b_2}\,\overline{c_1}\,\overline{c_2} c_4 d_1
\end{aligned}
$$

$$
\overline{Z} = a_1 \overline{b_1} + \overline{b_1} c_1 + \overline{a_2} b_2 + \overline{a_2} c_2 + b_3 \overline{d_1} + \overline{d_1} c_3 + b_4 \overline{c_4} + d_2 \overline{c_4}
$$

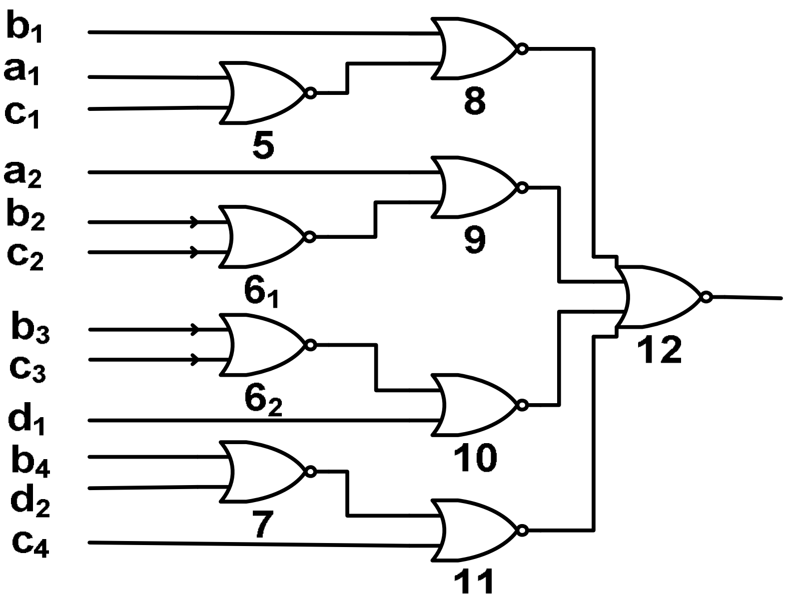

Рис.3. Еквівалентне дерево

Як можна бачити з наведених виразів, ЕНФ відрізняється від звичайної ДНФ тим, що надлишкові терми не видаляються. Кожен терм ЕНФ (ОЕНФ) представляє шлях в схемі від входів до виходу.

Кожна константна несправність еквівалентна фіксації деякої множини букв ЕНФ (ОЕНФ) значеннями 0 або 1. Так несправність ***6₀*** в схемі, що розглядається, може бути зафіксована будь-яким з наступних наборів значень змінних $b_2 c_2 b_3 c_3$: 0101, 0110, 0111, 1001, 1010, 1011, 1101, 1110, 1111.

Якщо несправність i може бути викрита, то завжди існує вхідний набор X такий, на якому $Z=0$, $Z^i=1$, або $\overline{Z}=0$, $\overline{Z^i}=1$, де Z^i - функція схеми з несправністю i. Разом з цим, завжди існує хоча б один терм k, у якому несправність i, по-перше, фіксує непусту множину букв Q_k значенням 1, по-друге, не фіксує жодної букви терма значенням 0. Такий терм називають одиничним відносно несправності i. Набор X, який викриває несправність i, по-перше, встановлює значення 1 всім буквам k-го терма, які не належать множині Q_k, по-друге, значення 0 – хоча б однієї букві з множини Q_k, і, по-третє, значення 0 – щонайменше однієї букві в кожному з решти термів ЕНФ (ОЕНФ). Виконання всіх умов представляється у формі логічного добутку

$$æ_k' \, æ_k'' \, \rho,$$

де $æ_k'$ – диз'юнкція інверсій всіх змінних, представлених буквами множини; Q_k, $æ_k''$ – кон'юнкція змінних *k*-го одиничного відносно *i* терма, які не належать Q_k, ρ – вираз, що описує умови, при яких решта термів ЕНФ дорівнює 0.

Приклад. Знайдемо методом ЕНФ тест несправності ***б₀*** в схемі контрприклада Шнейдера. Знайдений (якщо існує) вхідний набор *X* повинен забезпечувати умови, при яких $\overline{Z}=0,\ \overline{Z^i}=1$.

Несправність ***б₀*** фіксується набором 0101 значень змінних $b_2c_2b_3c_3$. Першим одиничним відносно несправності термом є терм $\overline{a_2c_2}$, в якому c_2 фіксується в 1, тобто $k=4,\ Q_k=\{c_2\}$. Для цього терма

$$æ_4' = \bar{c},\ æ_4'' = \bar{a},\ \rho = Z\,.$$

Замість Z у ЕНФ можна застосувати вираз для функції на виході, що досліджується, у мінімальній формі.

$$Z = abcd + \overline{abcd}$$

Одночасне виконання всіх умов дає тест несправності ***б₀***:

$$æ_4' \, æ_4'' \, \rho = \overline{ac}\,(abcd + \overline{abcd}) = \overline{abcd}\,.$$

Лема 3. Обчислювальна складність вирішення задачі побудови теста для пари технічних станів логічної мережі, яка має *m* вхідних полюсів, методом ЕНФ дорівнює 2^m.

Доказ. Доказ можна провести аналогічно доказу леми 2. Дійсно, пошук коренів рівняння $æ_k' \, æ_k'' \, \rho = 1$, представленого у КНФ, є *NP*-повною задачею, складність якої - 2^m, де *m* – кількість змінних у рівнянні. Таким чином, складність побудови теста методом ЕНФ також дорівнює 2^m. Лему доведено.

Метод розрізняючої функції

Задача побудови теста методом розрізняючої функції формулюється наступним чином: знайти вхідний набор *X*, що задовольняє щонайменше одному з *n* булевих рівнянь:

$$D_f\left[F_i(X)\right] = 1, \quad i = 1,2,\ldots,n\,,$$

де $D_f\left[F_i(X)\right]$ - розрізняюча функція, яка визначається наступним виразом

$$D_f\left[F_i(X)\right] = F_i(X) \oplus F_i^f(X),$$

$F_i(X),\ F_i^f(X)$ - функції *i*-го виходу схеми, відповідно, у справному стані і при несправності φ_f.

Метод розрізняючої функції дозволяє замінити процедуру безпосереднього вирішення системи *n* рівнянь рекурентною процедурою обчислення одного або декількох розріняючих вхідних наборів по логічній мережі. Основою цієї процедури є операція зворотної підстановки і обчислення розрізняючої функції виходу певного елемента по розрізняючим функціям його входів і функціям, які реалізують справні і несправні логічні елементи мережі.

При цьому використовуються наступні позначення:

$I_1, I_2, \ldots, I_m$ - вхідні змінні;

$F_1, F_2, \ldots, F_n$ - змінні виходів;

$C_1, C_2, \ldots, C_p$ - змінні елементів мережі;

$E_1, E_2, \ldots, E_p$ - логічні вирази функцій елементів мережі:

$E_i : C_i = \sum_k v_k^{s(k)}$ або $E_i : C_i = \prod_k V_k^{s(k)}$,

де $V_k^{s(k)}$ - вхідні або внутрішні змінні мережі, $s(k) = 0{,}1, V_k^o = \overline{V_k}, V_k^1 = V_k$;

$\varphi_1, \varphi_2, \ldots, \varphi_l$ - несправності;

$[C]^f$ - елементи, несправність яких є причиною появи φ_f ;

E_i^f - логічний вираз функції елемента $C_i \in [C]^f$;

$F_i^f(X), C_k^f(X)$ - змінні елементів мережі з несправністю φ_f , значення яких можуть відрізняться від значень F_i та C_k .

Алгоритм пошуку теста для викриття несправності φ_f по реакції на *i*-му виході, складається з наступних кроків.

1. $S = 0$. $W_S = D_f\left[F_i^f\right]$.

2. Вибрати у W_S перший ненульовий терм *T*.

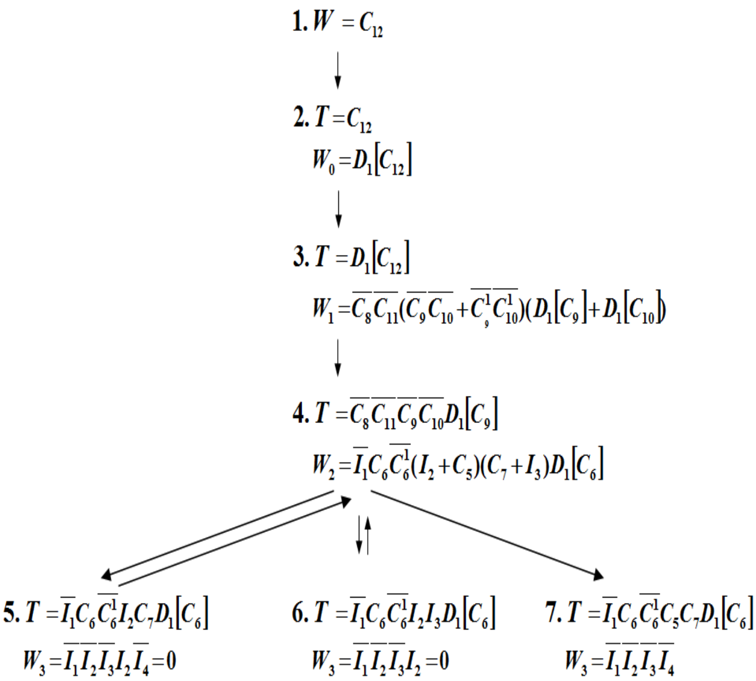

Рис.4. Метод розрізняючої функції

3. Обраний терм має вигляд $T = \left(\prod_n V_n^{s(n)}\right)\left(\prod_k C_k^{f,s(k)}\right) D_f\left[C_j\right]$, де $V_n^{s(n)}$ - змінні, значення яких не залежать від φ_f. Замінити $V_n^{s(n)}$ і $C_k^{f,s(k)}$ на праві частини логічних виразів E_n і E_k^f. Обчислити $D_f\left[C_j\right]$, замінивши C_j на E_j і E_j^f.

4. $S = S + 1$ W_S привласнити результат, отриманий при виконанні п.3.

5. Вибрати в W_S черговий ненульовий терм *T*. Якщо *T* складається тільки з вхідних змінних, то тест знайдено, інакше перейти до п.3. Якщо всі терми в W_S нульові, то $S = S - 1$. Якщо $S \neq 0$, то перейти до п.5, в іншому випадку несправність φ_f не може бути викрити по реакції на *i*-му виході мережі.

Приклад. Побудувати тест несправності ***6₀*** в схемі, наведеній на рис.1. Процес побудови в формі дерева показано на рис.4. Вершинам відповідають ненульові терми, які обрано в п.4 алгоритму. Послідовність дій вказана стрілками.

Лема 4. Обчислювальна складність методу розрізняючої функції дорівнює 2^m, де m – кількість входів схеми, для якої будується тест.

Доказ. Пошук теста методом розрізняючої функції полягає у вирішенні булевого рівняння $D_f[F_i(X)] = 1$. Алгоритм пошуку рішення вимагає завершити пошук при знаходженні першого ненульового терма T, складеного тільки з вхідних змінних I_k. Проте, у випадку, коли теста несправності φ_f не існує, буде необхідно перебрати всі терми T, які породжуються з $D_f[F_i(X)]$. При цьому складність рішення залежатиме від складності аналітичних перетворень розрізняючої функції $D_f[F_i(X)]$. Дану функцію можна виразити через вхідні змінні у КНФ. Таким чином, складність побудови теста методом розрізняючої функції дорівнює складності задачі про перевірку виконуваності КНФ. Лему доведено.

Метод фокусованого пошуку

Проведений аналіз методів вирішення задачі побудови тестів для комбінаційних схем дозволяє зробити висновок, що головною причиною їх високої складності є некерований перебір всіх можливих комбінацій сигналів, які забезпечують проявлення несправності на виході несправного елемента і транспортування викривленого сигналу до одного з виходів схеми. В [12] приведено результати теоретичних досліджень і дані експериментальної перевірки ефективності метода, призначеного для скорочення величини перебору в задачі побудови тестів.

Вважається доцільним доповнити приведений вище аналіз висновками стосовно складності побудови тестів методом фокусованого пошуку.

Лема 5. Обчислювальна складність вирішення задачі побудови теста для пари технічних станів логічної мережі методом фокусованого пошуку дорівнює 2^m.

Доказ. Скорочення перебору методом фокусованого пошуку виконується шляхом застосування *заборон* (запретов,-*рос*.), в яких накопичується інформація про тупикові вершини дерева призначення сигналів. Оскільки складність оцінюється для найгіршого випадку, то необхідно розглядати варіант рішення задачі, в якому жодна з породжених вершин не задовольняє умовам жодної із сформованих заборон. В цьому випадку складність дорівнює складності вирішення задачі призначення сигналів без скорочення перебору. При виконанні пошуку задачі про призначення сигналів кожна наступна вершина дерева формується як результат вибору чергової кон'юнкції з ДНФ літери, яка стоїть першою у комбінації сигналів попередньої вершини. У випадку, коли теста не існує, буде виконано перебір всіх кон'юнкцій літер, для яких виконуються підстановки. Тому можна стверджувати, що обсяг обчислень буде еквівалентним пошуку рішення шляхом підстановки в усі вершини, починаючи з кореневої, замість кожної літери всієї її функції в ДНФ. Результатом такої підстановки буде логічний вираз, що складається тільки з вхідних змінних. Очевидно, що складність наступних перетворень має таку ж оцінку, що і задача про перевірку виконуваності КНФ і дорівнює 2^m. Лему доведено.

Висновки

1. Всі відомі класичні методи побудови тестів базуються на некерованому переборі 2^m наборів значень вхідних сигналів. Жоден з цих методів не пропонує ніяких засобів скорочення перебору, тому їх низька ефективність для реальних схем не дозволяє надати перевагу жодному з них.

2. Для скорочення перебору при пошуку рішення в системах продукційного типу, до яких також відноситься задача побудови тестів, може бути застосований метод фокусованого пошуку.

3. Хоча оцінкою складності метода фокусованого пошуку, як і всіх інших методів, є величина 2^m, проведені експерименти показують, що наявність механізмів скорочення перебору дозволяє значно обмежити обсяг необхідних обчислень при побудові тестів. Отримані результати [12] підтверджують високу ефективність метода фокусованого пошуку і можуть розглядатися, як логічне обґрунтування вибору даного методу для розробки діагностичного забезпечення сучасної цифрової техніки.

Література:

1. Карибский В.В., Пархоменко П.П., Согомонян Е.С. Основы технической диагностики / Под ред. П.П.Пархоменко. -М.:Энергия,1976, -463с.

2. Бадулин С.С., Барнаулов Ю.М., Бердышев В.А. и др. Автоматизированное проектирование цифровых устройств / Под ред. С.С. Бадулина. -М.: Радио и связь, 1981, -240с.

3. Уткин А.А. Универсальный подход к построению тестов.-Сборник научных трудов ИТК АН БССР, Минск, 1981, с.5-16.

4. Компьютер и задачи выбора / Автор предисловия Ю.И. Журавлев. М.: Наука, 1989, -208с.

5. Иоффе М.И. Диагностирование логических схем. -М.: Наука, 1989, -159с.

6. Roth J.P., Bouvicins W.G., Schneider P.R. Programmed Algorithms to Computer Tests to Detect and Distinguish Between Failures in Logic Circuits. -IEEE Trans., 1967, v.EC-16, № 5,p.567-580.

7. Armstrong D.B. On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets. –IEEE Trans. on Electron. Comput., 1966, v.EC-15, p.66-73.

8. Sellers F.F., Hsiao M.Y., Bearnson L.W. Analyzing Errors with the Boolean Difference. -IEEE Trans. on Comput., 1986, v.C-17, p.676-683.

9. Armar V., Condulmari N. Diagnosis of Lage Combinational Networks. -IEEE Trans. on Electron. Comput.,1967, v.EC-16, № 10, p.675-680.

10. 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.

11. Гэри М., Джонсон Д. Вычислительные машины и трудноразрешимые задачи.-М.: Мир, 1982, 416с.

12. Куліков В.М. Підхід до побудови тестів перевірки цифрових пристроїв на надвеликих інтегральних схемах. // Інформаційні технології і безпека : сб.наук.пр. №1(1) / ІСЗЗІ НТУУ "КПІ". – Київ, 2011. – С. 83-92.