DOI: https://doi.org/10.33216/2222-3428-2025-28-2
УДК 004.94

Рязанцев О.І., Кардашук В.С. , Барбарук В.М., Кравцов С.В.

ДОСЛІДЖЕННЯ ПОБУДОВИ ТЕСТОВИХ ПОСЛІДОВНОСТЕЙ ДЛЯ ПЕРЕВІРКИ ФУНКЦІОНУВАННЯ ПЛІС

У статті проведено дослідження методів побудови тестів для перевірки функціонування ПЛІС. Застосування методів діагностики та тестування при їх практичній реалізацій дозволить краще зрозуміти їх можливості та обмеження, а також визначити оптимальний алгоритм для конкретних завдань забезпечення працездатності систем на електронних компонентах. Аналіз існуючих методів та алгоритмів побудови тестів для перевірки працездатності засобів цифрових систем, їх класифікація, застосування та порівняння дозволяє побудувати оптимальну структуру тестів, скоротити кількість перевірок та при програмній реалізації скоротити об'єм програмного коду. Здійснено огляд методів побудови тестів в системах діагностики комп'ютерних систем на базі ПЛІС. Відмічені основні принципи логічного моделювання: паралельність, подієвість, асинхронність, ієрархічність. Відзначені основні риси принципу подієвості. Запропонований алгоритм знаходження логічної моделі на основі вхідних послідовностей (наборів). Розглянуто логічну модель, що містить 4 входи та 1 вихід, метод зворотної імплікації з метою знаходження тестового впливу для реалізації принципу подієвості. Розглянута зворотна імплікація для цифрової схеми полягає в обчисленні множини вхідних значень сигналів, подаючи які на схему, можна отримати необхідні вихідні значення. При виконанні зворотної імплікації використовуються операції кубічного перетину вхідного вектору з кубічним вектором покриття елемента. Наведено приклад виконання зворотної імплікації для одиничного та нульового значення на виході схеми. Отримані набори для одиничного C^1 та нульового покриття C^0 використовуються для подачі на схему тестового впливу. Перевірки правильності функціонування для визначення одиничних та нульових значень на повному наборі вхідних впливів проведена в програмі Multisim 14.3. В подальшому результати дослідження можуть бути використані для перевірки функціонування вузлів та блоків систем на базі ПЛІС, побудові таблиць константних несправностей та алгоритмів дедуктивного моделювання, який є найпростішим неявним методом моделювання несправностей, орієнтованим на обробку схем вентильного логічного рівня або універсального базисів.

Ключові слова: *ПЛІС, тестова послідовність, моделювання, подієвість, зворотна імплікація, кубічне покриття.*

Актуальність дослідження. Цифрові системи на базі ПЛІС (програмованих логічних інтегральних схемах) розглядаються як модулі, що виконують задані функції в межах заданих булевих рівнянь, що описують кожен вихідний сигнал на основі вхідних та їх технічних характеристик – кількість вентилів, час затримки, кількість входів/виходів тощо. Всі вони мають внутрішню структуру, яка підлягає діагностичному тестуванню. Задача розробників створити таку послідовність тестових впливів, щоб мати змогу перевірити функціонування булевих рівнянь з мінімальною кількістю тестових впливів. Такі тестові впливи дозволяють визначити проблемні точки електронної схеми/вентиля та локалізувати дефектні елементи для електронних пристроїв, місця пайки та інші моменти, що можуть впливати на позитивний кінцевий результат перевірки [1].

Такий підхід до питання створення тестів для цифрових систем є запорукою для тривалої експлуатації електронного пристрою – ПЛІС, модулів з пам'яттю тощо.

Постановка проблеми. Тестування застосовується до апаратних засобів – мікропроцесорів, мікроконтролерів, модулів пам'яті, логічних елементів, матриць тощо. Дослідницька система у цьому випадку розглядається як «чорний» ящик» з n-входами та m- виходами Причини, які спонукають звертатись до проведення тестових діагностичних процедур – дефекти електронних компонентів, невірність монтажу, помилки проектування.

Вдало проведені процедури тестування є запорукою тривалої роботи електронного пристрою, мікросхеми, окремого модулю тощо. При подачі на систему/модуль/схему тестового впливу перевіряється реакція системи, з видачою результату перевірки – позитивний/негативний/відсутність збоїв тощо.

Зазвичай, при відсутності реального об'єкту на початковому етапі інструментальними засобами створюється її електронна модель. Наприклад використовуються VHDL, Multisim, Max+ Plus II, PLD Shell та ряд інших.

Аналіз останніх досліджень і публікацій. ПЛІС відносяться до складу комбінаційних пристроїв на рівні апаратних застосувань. На ПЛІС в цифрових системах покладаються функціональні завдання, що реалізуються відповідним заданими функціональними рівняннями. В процесі налаштування такі рівняння можуть бути змінені (перепрограмовані) під різноманітні рішення, а розміщення ПЛІС на технологічній панелі є доцільними в режимі налаштуванні цифрового пристрою (рис. 1) [2].

Рисунок 1 – ПЛІС фірми Altera

Структурна схема ПЛІС є універсальною і розглядається як «чорний ящик» n-входами та m-виходами. При такому підході користувачу можуть бути невідомі навіть функціональні рівняння, що запрограмовані.

Для прикладу, приведена структурна схема ПЛІС 22v10B фірми Atmel (рис. 2). Дана ПЛІС відноситься до складу простих та може розглядатись як базова, для освоєння більш складних по структурі та функціональним можливостям ПЛІС [3].

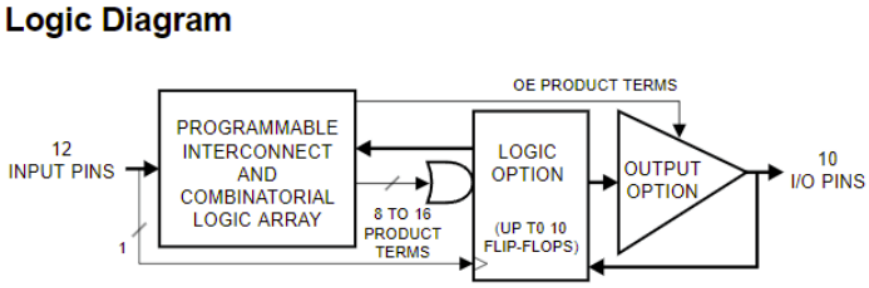


Рисунок 2 – Структура ПЛІС Atmel 22V10

У версії ПЛІС 22V10B міститься 12 входів і 10 виходів Відображена структура співставна зі структурою «чорного» ящика, де є 12 вхідних та 10 вихідних сигналів. Робота ПЛІС здійснюється у відповідності до вхідних та запрограмованих булевих функціональних вихідних сигналів. Функція виходу задається булевим рівнянням логічної функції до якої входять вхідні терми.

Слід зауважити, що довжина логічного рівняння може складатись з 7 доданків кожен з яких може містити до 7 термів. У випадку, коли необхідно збільшити кількість доданків, результуюче рівняння «заводиться» в матрицю та слугує в подальшому для участі у іншому рівнянні. Збільшуючи довжину рівняння, додатково втрачається один вихід матриці.

Тобто, користувачу, при зверненні до ПЛІС, необхідно описати тільки логічні рівняння та визначити виходи. Внутрішня «комутація» сигналів покладена на структуру ПЛІС. Необхідний вихідний логічний рівень 0 або 1, згідно рівняння, покладено на ПЛІС при її програмуванні.

Характерною особливістю підходу до проектування логічного тестування є врахування наступних принципів паралельності, подієвості, асинхронності та ієрархічності [4].

Подієвість у цьому переліку відіграє особливо важливу роль, оскільки відображає реакцію схемної частини на тестовий вплив. Подієвість, на основі переключення стану виходу, дозволяє визначити працездатність вузла схеми, що досліджується (рис. 3)

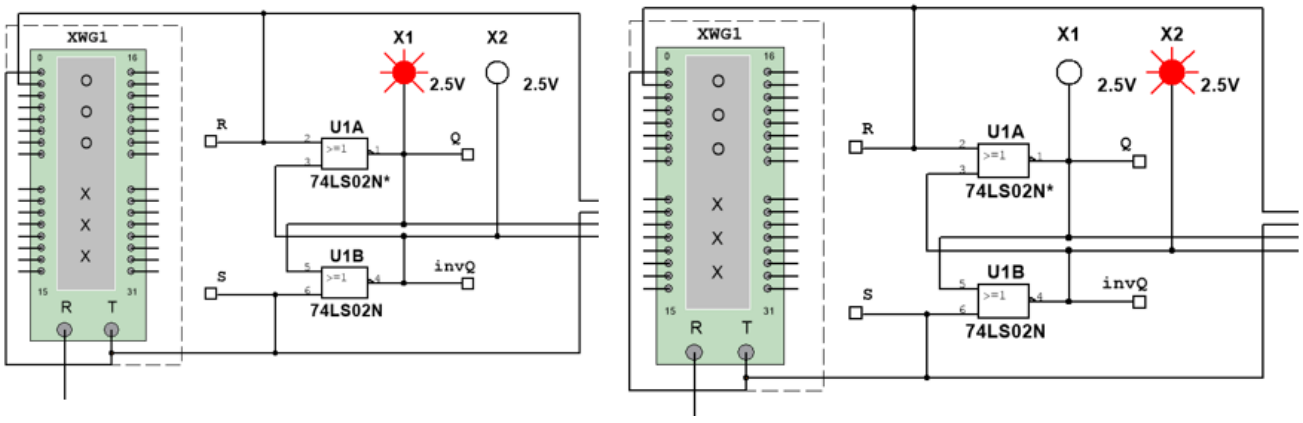


Рисунок 3 – Реалізація принципу подієвості

Задача розробників створити таку послідовність тестових впливів, щоб мати змогу перевірити функціонування модулю з мінімальною кількістю тестів.

Вирішення проблеми. Розглянемо логічну модель, що містить 4 входи та 1 вихід. Якщо логічна модель невідома, а присутні тільки значення, які приймаються на основі вхідних послідовностей (наборів), то в цьому випадку пропонується наступний алгоритм реалізації знаходження логічної моделі:

1. Створюємо перелік вхідних послідовностей та визначаємо значення, у яких функція приймає одиничне значення та нульове значення. Наприклад, функція F приймає одиничне значення на наборах 0, 1, 2, 4, 5, 8, 10, 11 та 15. На інших наборах функція F приймає нульове значення.

2. За допомогою карти Карно 4x4 [5] проводимо мінімізацію функції в ДНФ (диз`юнктивній нормальній формі).

X_1X_2	X_3X_4			
	00	01	11	10
00	1	1	0	1
01	1	1	0	0
11	0	0	1	0
10	1	0	1	1

3. В результаті мінімізації отримуємо логічне рівняння для ДНФ:

$$\boldsymbol{F}= / X_2 \cdot / X_4 + / X_1 \cdot / X_3 + X_1 \cdot X_3 \cdot X_4 \qquad (1)$$

Знак «/» перед сигналом свідчить про те, що сигнал визначається в інверсному коді (лог. 0). Замінюючи сигнали їх логічними еквівалентами, отримуємо тестову послідовність для перевірки функції F:

$$F= X0X0 + \ 0X0 \ + 1X11, \text{ де } X = 0 \text{ або } 1. \qquad (2)$$

Повне покриття та тестовий набір для діагностики має наступний вигляд:

Повне покриття				Тестовий набір			
X_1	X_2	X_3	X_4	X_1	X_2	X_3	X_4
X	0	X	0	0	0	0	0
				0	0	1	0
				1	0	0	0
				1	0	1	0
0	X	0	X	0	0	0	0
				0	0	0	1
				0	1	0	0
				0	1	0	1
1	X	1	1	1	0	1	1
				1	1	1	1

4. Побудувавши в програмі Multisim Education 14.3 [6] схемну модель, перевіримо функціонування отриманої моделі функції *F* для ДНФ [7], використавши (1), (рис. 4).

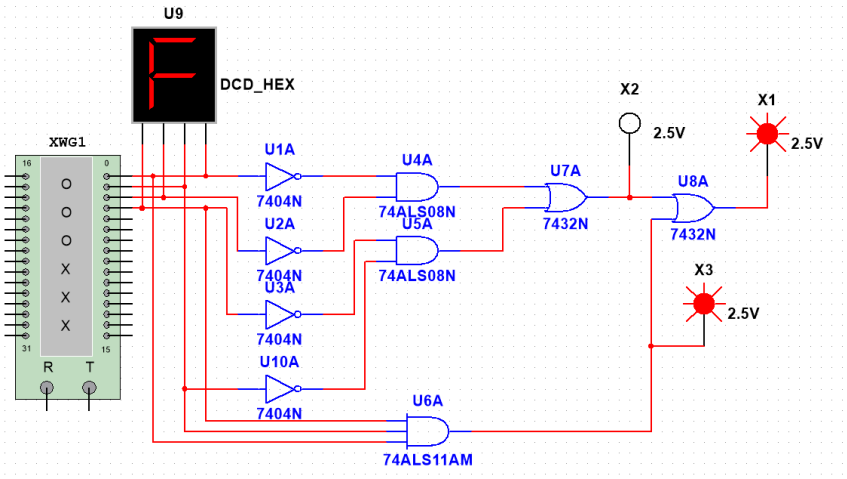


Рисунок 4 – Схемна модель функції *F* для ДНФ

При проведенні відповідних тестових процедур відсутні труднощі, пов'язані зі впливом коректності відображення моделі, що може викликати розбіжності в оцінці кінцевого результату, неповноти тесту. Що стосується часової затримки проходження сигналів через вентилі, то, наприклад, для ПЛІС часова затримка на вентилі складає 5 нс і є однаковою для всіх вентилів [3].

Практично всі нові створені компоненти потребують початкової діагностики та створення моделей.

При такому підході можна на етапі проектування виявити велику кількість неврахованих апаратних збоїв, дефектів тощо.

Тому, на передній план тут висуваються вимоги повноти тесту. Тестові послідовності (впливи) однозначно прив'язані до конкретної апаратури.

Зазвичай, такі процедури проводяться на початковому етапі роботи пристрою/модулю з видачою відповідних повідомлень про завершення проходження тесту – тест пам'яті, внутрішніх регістрів, виконання команд груп різноманітних операцій, у випадку тестування мікропроцесора/мікроконтролера тощо. Для перевірки функціонування моделі в КНФ, перетворимо (1), згідно правил алгебри логіки та законів де Моргана [8] для КНФ.

Спростивши вираз, отримаємо кінцеву формулу в КНФ для побудови моделі:

$$F = (/X_1+/X_2+X_3)(/X_1+/X_2+X_4)(X_1+/X_2+/X_3)(/X_2+/X_3+X_4)(/X_1+X_3+/X_4)(X_1+/X_3+/X_4) \qquad (2)$$

Представлена схемна модель для КНФ, згідно (2), зображена на рисунку 5.

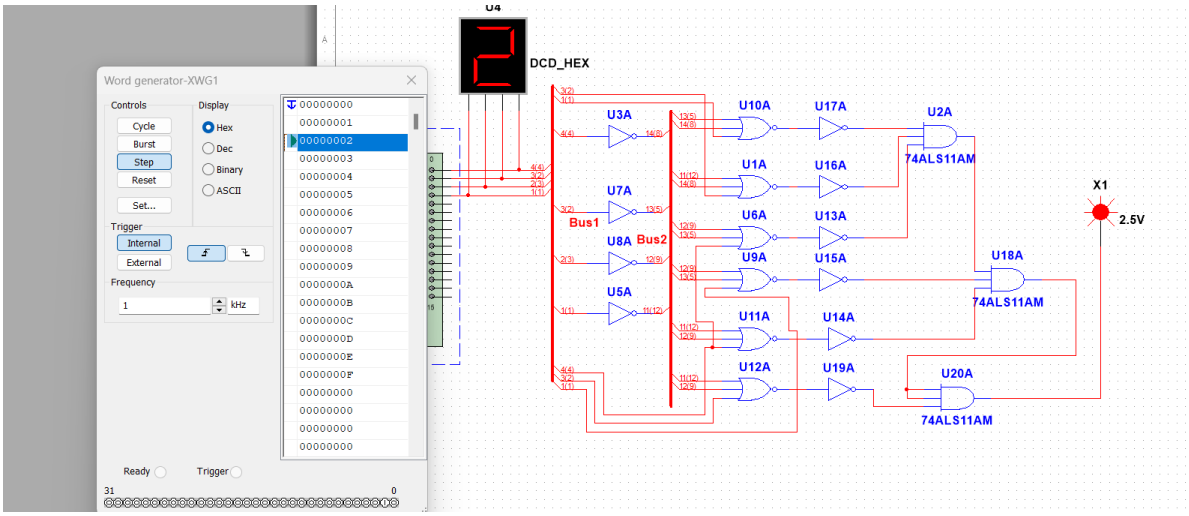
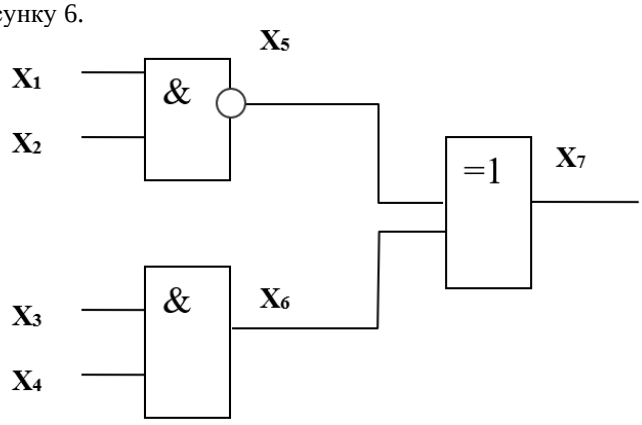


Рисунок 5 – Схемна модель функції *F* для КНФ

У порівнянні з моделлю для ДНФ (див. рис. 4), модель для КНФ містить більше логічних елементів. Вхідні набори в генераторі слів будуть такі ж самі, як і в попередньому випадку. Моделювання показало тотожність роботи моделей, як для ДНФ, так і для КНФ. Отже, моделювання виконано вірно, що підтверджують результати перевірки.

Ще одним з методів знаходження тестового впливу для реалізації принципу подієвості є метод зворотної імплікації. Зворотна імплікація для цифрової схеми полягає в обчисленні множини вхідних значень сигналів, подаючи які на схему, можна отримати необхідні вихідні значення.

Покажемо виконання зворотної імплікації [9] для комбінаційної схеми для одиничного і нульового значень на виході X_7 схеми, наведеної рисунку 6.



Риунок 6 – Комбінаційна схема

При виконанні зворотної імплікації використовуються операції кубічного перетину вхідного вектору з кубічним вектором покриття елемента [4]. Приклад виконання зворотної імплікації для одиничного та нульового значення на виході схеми X_7:

X_1	X_2	X_3	X_4	X_5	X_6	X_7	Примітки
x	x	x	x	x	x	1	T – Вихідний вектор довизначення лінії 7
				0	1	1	C_1
				1	0	1	C_2
x	x	x	x	0	1	1	$T_1=T \cap C_1$
x	x	x	x	1	0	1	$T_2=T \cap C_2$
		0	x		0		C_3
		x	0		0		C_4
		1	1		1		C_5
x	x	0	x	0	Ø		$T_3=T_1 \cap C_3$
x	x	x	0	0	Ø		$T_4=T_1 \cap C_4$
x	x	1	1	0	1	1	$T_5=T_1 \cap C_5$
x	x	0	x	1	0	1	$T_6=T_2 \cap C_3$
x	x	x	0	1	0	1	$T_7=T_2 \cap C_4$
x	x	1	1	1	Ø		$T_8=T_2 \cap C_5$
0	x			1			C_6
x	0			1			C_7
1	1			0			C_8
0	x	1	1	Ø			$T_9=T_5 \cap C_6$
Остаточні куби							
x	0	1	1	Ø			$T_{10}=T_5 \cap C_7$
1	**1**	**1**	**1**	**0**	**1**	**1**	$T_{11}=T_5 \cap C_8$
0	**x**	**0**	**x**	**1**	**0**	**1**	$T_{12}=T_6 \cap C_6$
x	**0**	**0**	**x**	**1**	**0**	**1**	$T_{13}=T_6 \cap C_7$
1	1	0	x	Ø			$T_{14}=T_6 \cap C_8$
0	**x**	**x**	**0**	**1**	**0**	**1**	$T_{15}=T_7 \cap C_6$
x	**0**	**x**	**0**	**1**	**0**	**1**	$T_{16}=T_7 \cap C_7$

X_1	X_2	X_3	X_4	X_5	X_6	X_7	Примітки
x	x	x	x	x	x	0	T – Вихідний вектор довизначення лінії 7
				0	0	0	C_1
				1	1	0	C_2
x	x	x	x	0	0	0	$T_1=T \cap C_1$
x	x	x	x	1	1	0	$T_2=T \cap C_2$
		0	x		0		C_3
		x	0		0		C_4
		1	1		1		C_5
x	x	0	x	0	0	0	$T_3=T_1 \cap C_3$
x	x	x	0	0	0	0	$T_4=T_1 \cap C_4$
x	x	1	1	Ø			$T_5=T_1 \cap C_5$
x	x	0	x	Ø			$T_6=T_2 \cap C_3$
x	x	x	0	Ø			$T_7=T_2 \cap C_4$
x	x	1	1	1	1	0	$T_8=T_2 \cap C_5$
0	x			1			C_6
x	0			1			C_7
1	1			0			C_8
0	x	0	x	Ø			$T_9=T_3 \cap C_6$
Остаточні куби							
x	0	0	x	Ø			$T_{10}=T_3 \cap C_7$
1	**1**	**0**	**x**	**0**	**0**	**0**	$T_{11}=T_3 \cap C_8$
0	x	x	0	Ø			$T_{12}=T_4 \cap C_6$
x	0	x	0	Ø			$T_{13}=T_4 \cap C_7$
1	**1**	**x**	**0**	**0**	**0**	**0**	$T_{14}=T_4 \cap C_8$
0	**x**	**1**	**1**	**1**	**1**	**0**	$T_{15}=T_8 \cap C_6$
x	**0**	**1**	**1**	**1**	**1**	**0**	$T_{16}=T_8 \cap C_7$
1	1	1	1	Ø			$T_{17}=T_8 \cap C_8$

Отримані остаточні куби, за виключенням пустих множин, для одиничного C^1 та нульового покриття C^0 використовуються для подачі на схему тестового впливу:

Тестовий набір C^1	Тестовий набір C^0
1111	110X
0X0X	11X0
X00X	0X11
0XX0	X011
X0X0	

Математична модель комбінаційної схеми (див. рис. 4), для якої формується тестовий вплив, описується рівнянням:

$$F= /(X_1 \bullet X_2) \oplus (X_3 \bullet X_4) \quad (3)$$

Для отримання рівняння в ДНФ скористаємось законами алгебри логіки та законами де Моргана. В результаті перетворення отримаємо наступну функцію:

$$F = /(X_1X_2) \oplus (X_3X_4) = (/(X_1X_2) + X_3X_4) + (//(X_1X_2) + /(X_3X_4)) = (/X_1 + /X_2 + X_3X_4) \bullet (X_1X_2 + /X_3 \vee /X_4) = \\ \cancel{/X_1X_1X_2} + \cancel{X_1X_2/X_2} + X_1X_2X_3X_4 + /X_1/X_3 + /X_2/X_3 + \cancel{X_3/X_3X_4} + \cancel{X_3X_4/X_4} + /X_1/X_4 + /X_2/X_4 + \cancel{X_3X_4/X_4} = \\ = X_1X_2X_3X_4 + /X_1/X_3 + /X_2/X_3 + /X_1/X_4 + /X_2/X_4 \quad (4)$$

Остаточне одиничне покриття (C^1) співпадає з отриманим в результаті зворотної імплікації для одиничного

значення на виході схеми.

Для отримання нульового покриття (C^0), з використанням математичної моделі, здійснюється приведенням рівняння до КНФ.

На завершальному етапі перевірки правильності функціонування, для визначення одиничних та нульових значень на повному наборі вхідних впливів, створимо схемну модель в програмі Multisim 14.3 (рис. 7).

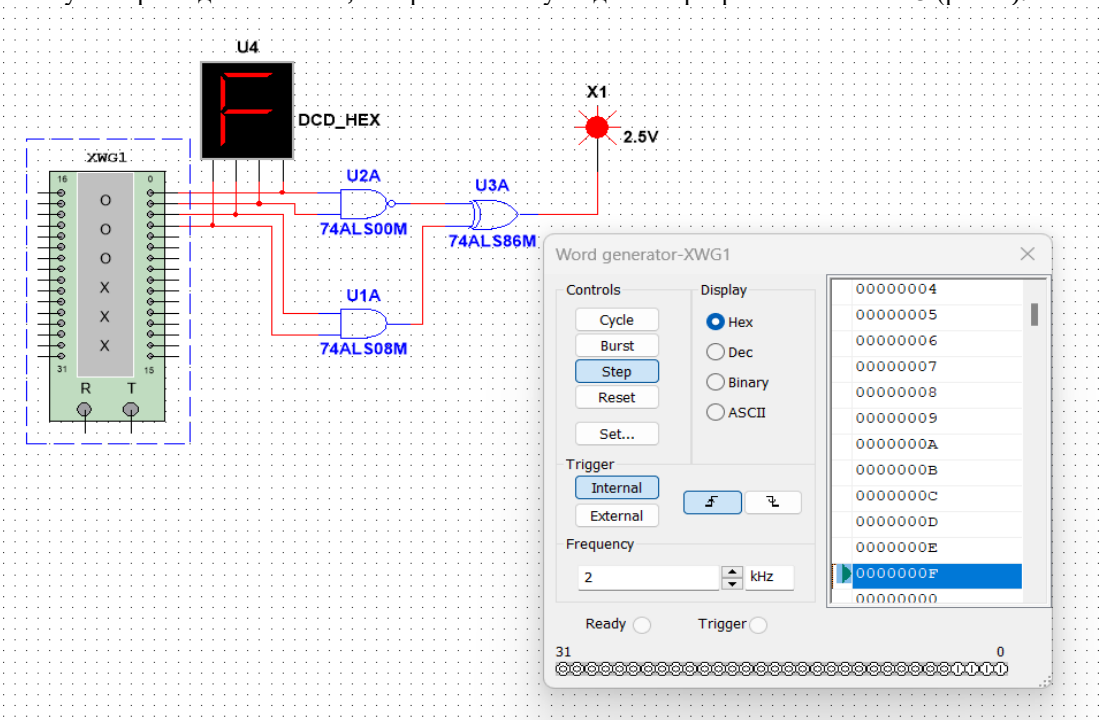


Рисунок 7 – Схемна модель

Функція *F* приймає одиничне значення на 0, 1, 2, 4, 5, 6, 8, 9, 10 та 15 вхідних наборах. На інших вхідних наборах – нульове значення.

На базі визначених значень одиничних наборів побудується карта Карно та проводиться мінімізація для отримання ДНФ. В результаті мінімізації отримуємо 5 склейок:

$$F= /X_1/X_3 + /X_2/X_3 + /X_1x_4 + /X_2/X_4 + X_1X_2X_3X_4 \quad (5)$$

Приводячи мінімізовану формулу ДНФ до кубічного виразу, отримаємо наступні куби одиничного покриття:

$$0X0X,\ X00X,\ 0XX0,\ X0X0,\ 1111 \quad (6)$$

Для перевірки правильності отриманих кубів нульового покриття будується карта Карно і здійснюється мінімізація функції (по нулям) для отримання КНФ.

Наочність Карти Карно, завдяки простоті, має незаперечні переваги порівняно з іншими методами мінімізації з кількістю змінних до шести.

Висновки. Здійснено огляд методів побудови тестів в системах діагностики комп'ютерних систем на базі ПЛІС. Відмічені основні принципи логічного моделювання: паралельність, подієвість, асинхронність, ієрархічність. Відзначені основні риси принципу подієвості.

Запропонований алгоритм знаходження логічної моделі на основі вхідних послідовностей (наборів). Розглянуто логічну модель, що містить 4 входи та 1 вихід, метод зворотної імплікації з метою знаходження тестового впливу для реалізації принципу подієвості. Зворотна імплікація для цифрової схеми полягає в обчисленні множини вхідних значень сигналів, подаючи які на схему, можна отримати необхідні вихідні значення.

При виконанні зворотної імплікації використовуються операції кубічного перетину вхідного вектору з кубічним вектором покриття елемента. Наведено приклад виконання зворотної імплікації для одиничного та нульового значення на виході схеми. Отримані набори для одиничного C^1 та нульового покриття C^0 використовуються для подачі на схему тестового впливу. Перевірки правильності функціонування, для визначення одиничних та нульових значень на повному наборі вхідних впливів, проведена в програмі Multisim 14.3.

В подальшому результати дослідження можуть бути використані для перевірки функціонування вузлів та блоків систем на базі ПЛІС, побудові таблиць константних несправностей та алгоритмів дедуктивного моделювання, який є найпростішим неявним методом моделювання несправностей, орієнтованим на обробку схем вентильного

рівня логічного (І, АБО, НІ) або універсального базисів [10].

Дослідження можуть бути використані для реалізації принципу структурного тестування, який полягає в транспортуванні несправності по активізованому шляху поданої на вхід елемента для виявлення її на одному із зовнішніх виходів схеми активізованим шляхом. В подальшому робота може бути використана для розгляду процедури отримання таблиць несправностей логічних елементів та перехід від суміщеної таблиці несправності до класичної для класів еквівалентних дефектів.

Розглянуті моделі цифрових пристроїв, що застосовуються при моделюванні та визначені рівні сигналів в компілятивному та інтерпретативному методі моделювання

Здійснено дослідження методів побудови тестів з використанням інтерпретативного методу моделювання цифрових схем з застосуванням структурного тестування.

Зазначено, що у задачах діагностики логічне моделювання застосовують в основному для вирішення двох класів завдань: обчислення реакцій схеми на вхідні дії в справному стані і обчислення реакцій схеми при наявності в схемі певних несправностей.

Література

1. Алсуф'єв Б. Б., Кардашук В.С. Методи побудови тестів в системах діагностики комп'ютерних систем Електронний ресурс. Режим доступу http://moodle2.snu.edu.ua/pluginfile.php/105057/mod_resource/content/1/Алсуфьев%20П3.pdf (дата доступу 01.02.2025).
2. Стешенко В.Б. ПЛИС фирмы ALTERA. Проектирование устройств обработки сигналов. Електронний ресурс. Режим доступу https://www.kafelec.nau.edu.ua/Materialu/PLD%20ALTERA.%20Proekt_ustr_obrob_signalov_1.pdf (дата доступу 02.02.2025).
3. AT22V10B Datasheet (PDF) - ATMEL Corporation. Електронний ресурс. Режим доступу https://www.alldatasheet.com/datasheet-pdf/pdf/129444/ATMEL/AT22V10B.html (дата доступу 01.02.2025).
4. Кардашук В.С. Моделювання несправностей в цифрових пристроях. Вісник Східноукраїнського національного університету імені Володимира Даля, № 8 (238), 2017. – С. 36-38.
5. Застосування карт Карно для мінімізації логічних функцій. Електронний ресурс. Режим доступу https://epa.kpi.ua/wp-content/Buryan/АТПУ-1.%20Лекція%204.pdf (дата доступу 03.02.2025).
6. Multisim. Електронний ресурс. Режим доступу https://www.ni.com/en/support/downloads/software-products/download.multisim.html?srsltid=AfmBOoqzA03qTea_OiImT0ksT-VavRcqzC_a1zvffHTgCzfGQCIHGByw#452133 (дата доступу 03.02.2025).
7. Алгоритм перетворення формули в КНФ і ДНФ. Електронний ресурс. Режим доступу https://stud.com.ua/171473/prirodoznavstvo/algoritm_peretvorennya_formuli#google_vignette (дата доступу 01.01.2025).
8. Основні закони алгебри логіки та їх використання для подання одних функцій логіки через інші. Електронний ресурс. Режим доступу https://web.posibnyky.vntu.edu.ua/fitki/6kondratenko_komp_praktikum_matlog/12..htm (дата доступу 03.02.2025).
9. Математична логіка та теорія алгоритмів. Електронний ресурс. Режим доступу https://ela.kpi.ua/server/api/core/bitstreams/90cfd1e3-1a98-4370-95f3-bd1d677d0e7c/content (дата доступу 01.02.2025).
10. Касабуцька І.В., Кардашук В.С. Методи вдосконалення тестування цифрових схем. Вісник Східноукраїнського національного університету імені Володимира Даля, № 5 (247), 2018. – С. 61-68.

References

1. Alsoufyev B. B., Kardashuk V. S. Methods of constructing tests in computer system diagnostics systems Electronic resource. Access mode http://moodle2.snu.edu.ua/pluginfile.php/105057/mod_resource/content/1/Алсуфьев%20П3.pdf (access date 01.02.2025).
2. Steshenko V. B. FPGA of ALTERA company. Design of signal processing devices. Electronic resource. Access mode https://www.kafelec.nau.edu.ua/Materialu/PLD%20ALTERA.%20Proekt_ustr_obrob_signalov_1.pdf (access date 02.02.2025).
3. AT22V10B Datasheet (PDF) - ATMEL Corporation. Electronic resource. Access mode https://www.alldatasheet.com/datasheet-pdf/pdf/129444/ATMEL/AT22V10B.html (access date 01.02.2025).
4. Kardashuk V.S. Modeling of faults in digital devices. Bulletin of the Volodymyr Dahl East Ukrainian National University, No. 8 (238), 2017. – P. 36-38.
5. Application of Karnaugh maps for minimizing logical functions. Electronic resource. Access mode https://epa.kpi.ua/wp-content/Buryan/АТПУ-1.%20Лекция%204.pdf (access date 03.02.2025).
6. Multisim. Electronic resource. Access mode https://www.ni.com/en/support/downloads/software-products/download.multisim.html?srsltid=AfmBOoqzA03qTea_OiImT0ksT-VavRcqzC_a1zvffHTgCzfGQCIHGByw#452133 (access date 03.02.2025).
7. Algorithm for converting a formula into CNF and DNF. Electronic resource. Access mode https://stud.com.ua/171473/prirodoznavstvo/algoritm_peretvorennya_formuli#google_vignette (access date 01.02.2025).
8. Basic laws of the algebra of logic and their use for representing some logic functions through others. Electronic resource. Access mode https://web.posibnyky.vntu.edu.ua/fitki/6kondratenko_komp_praktikum_matlog/12.htm (access date 03.02.2025).
9. Mathematical logic and theory of algorithms. Electronic resource. Access mode https://ela.kpi.ua/server/api/core/bitstreams/90cfd1e3-1a98-4370-95f3-bd1d677d0e7c/content (access date 01.02.2025).
10. Kasabutska I.V., Kardashuk V.S. Methods for improving testing of digital circuits. Bulletin of the Volodymyr Dahl East Ukrainian National University, No. 5 (247), 2018. – P. 61-68.

The article studies the methods of constructing tests for checking the functioning of PLD. The application of diagnostic and testing methods in their practical implementation will allow a better understanding of their capabilities and limitations, as well as determining the optimal algorithm for specific tasks of ensuring the operability of systems based on electronic components. Analysis of existing methods and algorithms for constructing tests for checking the operability of digital system tools, their classification, application and comparison allows building the optimal structure of tests, reducing the number of checks and, in the case of software implementation, reducing the volume of program code. A review of methods for constructing tests in diagnostic systems for computer systems based on PLD is carried out. The

main principles of logical modeling are noted: parallelism, event-based, asynchrony, hierarchical. The main features of the event-based principle are noted. An algorithm for finding a logical model based on input sequences (sets) is proposed. A logical model containing 4 inputs and 1 output is considered, the method of reverse implication is used to find a test effect for the implementation of the event principle. Reverse implication for a digital circuit consists in calculating a set of input signal values, applying which to the circuit, one can obtain the required output values. When performing reverse implication, the operations of cubic intersection of the input vector with the cubic element coverage vector are used. An example of performing reverse implication for a single and zero value at the circuit output is given. The obtained sets for a single C^1 and zero coverage C^0 are used to apply a test effect to the circuit. Checks of correct functioning, to determine single and zero values on a full set of input effects, were carried out in the Multisim 14.3 program. In the future, the results of the study can be used to check the functioning of nodes and blocks of PLD-based systems, build tables of constant faults and algorithms for deductive modeling, which is the simplest implicit method of modeling faults, focused on processing gate logic level or universal basis circuits.
Keywords: *PLD, test sequence, modeling, event-based, back implication, cubic coverage.*

Рязанцев О.І. – професор, завідувач кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету ім. В. Даля, **ryazancev@snu.edu.ua**
Кардашук В.С. – доцент кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету ім. В. Даля, **kardashuk@snu.edu.ua**
Барбарук В.М. – доцент кафедри інформаційних систем та технологій Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», barbaruk.viktor@gmail.com
Кравцов С.В. – аспірант кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету ім. В. Даля**, asp-122-22-434@snu.edu.ua**