

**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

Кафедра спеціалізованих комп'ютерних систем

МЕТОДИЧНІ ВКАЗІВКИ

до практичних занять, лабораторних робіт і курсового проєктування

з дисциплін

**«ТЕХНІЧНА ДІАГНОСТИКА ТА ТЕСТОПРИДАТНІСТЬ ЦИФРОВИХ
ПРИСТРОЇВ, КОМП'ЮТЕРНИХ СИСТЕМ, МЕРЕЖ ТА**

ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»,

**«ДІАГНОСТУВАННЯ ТА ТЕСТУВАННЯ КОМП'ЮТЕРНИХ
СИСТЕМ»**

Харків 2024

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри спеціалізованих комп'ютерних систем 12 лютого 2024 р., протокол № 8.

Методичні вказівки призначені для здобувачів вищої освіти першого (бакалаврського) рівня всіх форм навчання спеціальностей 123 «Комп'ютерна інженерія» і 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка», які вивчають дисципліни «Технічна діагностика та тестопридатність цифрових пристроїв, комп'ютерних систем, мереж та програмного забезпечення», «Діагностування та тестування комп'ютерних систем».

Укладач

доц. Л. А. Клименко

Рецензент

доц. А. О. Єлізаренко

ЗМІСТ

Вступ	4
1 Аналіз тестопридатності за методом Camelot	5
2 Проєктування тестопридатних цифрових автоматів методом сканованого шляху	15
3 Сигнатурні аналізатори в системах вбудованого самотестування	23
3.1 Сигнатурні аналізатори з внутрішніми елементами XOR у системах вбудованого самотестування	23
3.2 Сигнатурні аналізатори з зовнішніми елементами XOR у системах вбудованого самотестування	25
4 Забезпечення С-тестованості і побудова тесту для систолічних матриць комбінаційних елементів	30
5 Кубічне числення. Моделі елементів та основні операції	40
6 Асинхронне двійкове моделювання	46
7 Дедуктивне моделювання несправностей	51
Список літератури	58

ВСТУП

Діагностування цифрових електронних схем на рівні кристала, друкованої плати або системи є дорогим процесом. Вартість тестованості змінюється від 10 до 60 % і більше вартості повного життєвого циклу виробу залежно від складності виробу і позиції виробника, якому необхідно шукати кошти для фінансування цього процесу. Традиційно етапи логічного проектування виробу і синтезу тестового забезпечення для нього були поділені так: проектування передувало етапу синтезу тестів, а фахівців з тестової діагностики не залучали до процесу проектування.

Сьогодні розробники і діагности визнають, що для випуску на ринок виробів прийнятної якості на етапі виробництва їх слід адекватно протестувати. Для цього необхідно забезпечити тестопридатність виробів, починаючи з найбільш ранніх етапів проектування з залученням фахівців із тестової діагностики. Це поширюється на всі рівні проектування – від окремого кристала до кінцевої цифрової системи.

Технічна діагностика і тестопридатність цифрових пристроїв, комп'ютерних систем, мереж і програмного забезпечення є основним чинником, що спричиняє стрімкий розвиток залізничного транспорту та інфраструктури. У той час, коли потреби в тестопридатності та діагностиці збільшуються, з'являється можливість використання альтернативних методів і засобів тестопридатності та діагностики в комп'ютерних системах.

Вивчаючи цей курс, здобувачі не тільки зрозуміють основні принципи запобігання пошкодженню дорогого обладнання завдяки технічній діагностиці, контролю, але і отримають навички з тестопридатного проектування, отримання комп'ютерних систем і мереж та ефективного використання, вивчать норми державної політики щодо питання впровадження нових світових методів і засобів тестопридатності та діагностики цифрових пристроїв, комп'ютерних систем, мереж і програмного забезпечення.

1 АНАЛІЗ ТЕСТОПРИДАТНОСТІ ЗА МЕТОДОМ CAMELOT

Мета: набути навичок з оцінювання тестопридатності цифрових схем за методом Camelot, визначити керованість, спостережність і тестоприданість вузлів цифрових схем.

Завдання

Визначити керованість, спостережність і тестопридатність вузлів схеми; збільшити тестопридатність вузла з найнижчим її значенням, використовуючи заходи для покращення тестопридатності. Зробити повторні розрахунки показників тестопридатності.

Приклади виконання завдань

1 Визначити керованість, спостережність і тестопридатність вузлів схеми, зображеної на рисунку 1.1; збільшити тестопридатність вузла з найнижчим її значенням.

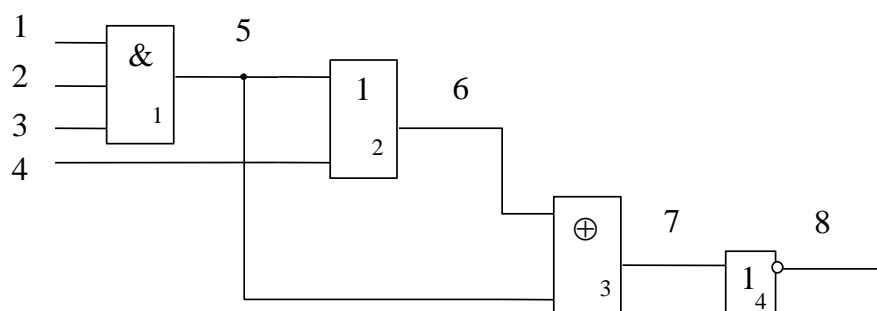


Рисунок 1.1

У методі Camelot вузли – еквіпотенціальні лінії (пронумеровані на лініях), пристрої – елементи схеми (всередині елемента).

Спочатку визначаємо **керованість** для усіх вузлів за формулою

$$CY(\text{вихідного вузла}) = KCY f (CY(\text{вхідних вузлів})), \quad (1.1)$$

де KCY – коефіцієнт передавання керованості пристрою;

f – середнє арифметичне керованості вхідних вузлів.

$$KCY = 1 - \frac{|N(0) - N(1)|}{|N(0) + N(1)|}, \quad (1.2)$$

де $N(0)$ – кількість усіх способів установлення логічного «0» на виході пристрою;

$N(1)$ – кількість усіх способів установлення логічної «1» на виході пристрою.

$$CY(1) = CY(2) = CY(3) = CY(4) = 1,$$

$$CY(5) = KCY(1)(CY(1)+CY(2)+CY(3))/3 = 0.25,$$

$$CY(6) = KCY(2)(CY(4)+CY(5))/2 = 0.3125,$$

$$CY(7) = KCY(3)(CY(5)+CY(6))/2 = 0.281,$$

$$CY(8) = KCY(4)CY(7) = 0.281.$$

Наступний крок – визначення **спостережності** за формулою

$$OY(I-Q) = OY(C-Q)KOY(I-C)g(CY(\text{активуючих входів})), \quad (1.3)$$

де $OY(I-Q)$ – спостережність вузла I у вузлі Q ;

$OY(C-Q)$ – спостережність вузла C у вузлі Q ;

$KOY(I-C)$ – коефіцієнт передавання спостережності від входу I до виходу C ; g – середнє арифметичне керованості входів, що активізують.

$$KOY(I-Q) = \frac{N(PDC:I-Q)}{N(PDC:I-Q) + N(NPDC:I-Q)}, \quad (1.4)$$

де $N(PDC:I-Q)$ – кількість одновимірних неполяризованих D-кубів, що активізують шлях I-Q;

$N(NPDC:I-Q)$ – кількість одновимірних неполяризованих D-кубів, що замикають і блокують активізацію шляху I-Q.

У розглянутому варіанті схеми функції елементів симетричні відносно їхніх входів, тому допускають запис КОУ (номер елемента) замість КОУ(I-Q).

$$KOY(1-5) = KOY(2-5) = KOY(3-5) = KOY(1) = 1/(1+3) = 0.25,$$

$$KOY(5-6) = KOY(4-6) = KOY(2) = 1/(1+1) = 0.5,$$

$$KOY(5-7) = KOY(6-7) = KOY(3) = 1/(1+0) = 1,$$

$$KOY(7-8) = KOY(4) = 1.$$

$$OY(8-8) = 1.$$

$$OY(7-8) = KOY(4)*OY(8-8) = 1.$$

$$OY(6-8) = KOY(3)*OY(7-8)*CY(5) = 0.25 \text{ (шлях 6-7-8).}$$

$OY(5-8) = KOY(3)*OY(7-8)*CY(6) = 0.3125$ (обираємо шлях 5-7-8, бо він коротше шляху 5-6-7-8).

$$OY(4-8) = KOY(2)*OY(6-8)*CY(5) = 0.03125.$$

$$OY(1-8) = OY(2-8) = OY(3-8) = KOY(1)*OY(5-8)*(CY(2)+CY(3))/2 = 0.078125.$$

Наступний крок – визначення **тестопридатності** за формулою

$$TY \text{ вузла} = CY \text{ вузла} OY \text{ вузла}. \quad (1.5)$$

$$TY(1) = CY(1) * OY(1) = 0.078125,$$

$$TY(2) = 0.078125,$$

$$TY(3) = 0.078125,$$

$$TY(4) = 0.03125,$$

$$TY(5) = 0.078125,$$

$$TY(6) = 0.078125, T,$$

$$Y(7) = 0.281, T,$$

$$Y(8) = 0.281.$$

Для полегшення інтерпретації результатів визначаємо CY^* , OY^* , TY^* , $CY^* = \sqrt[8]{CY}$, $OY^* = \sqrt[8]{OY}$, $TY^* = \sqrt[8]{TY}$. Значення показників до модифікації зазначені в таблиці 1.1.

Таблиця 1.1 – Значення показників до модифікації

Номер вузла	CY^*	OY^*	TY^*
1	1	0.727	0.727
2	1	0.727	0.727
3	1	0.727	0.727
4	1	0.648	0.648
5	0.841	0.865	0.727
6	0.865	0.841	0.727
7	0.853	1	0.853
8	0.853	1	0.853

Тестопридатність схеми розраховують за формулою

$$TY \text{ схеми} = (TY \text{ вузлів}) / \text{кількість вузлів.} \quad (1.6)$$

Найнижча тестопридатність у четвертому вузлі. Щоб збільшити тестопридатність вузла, необхідно насамперед збільшити його керованість. Оскільки вузол 4 має максимально можливу керованість, необхідно вжити заходів для покращення його спостережності. Якщо вузол 4 зробити таким, що спостерігає безпосередньо, спостережність інших вузлів не збільшиться, а тестопридатність усієї схеми, отже, збільшиться на 4%. Краще зробити спостерігачем безпосередньо вузол 6, як показано на рисунку 1.2.

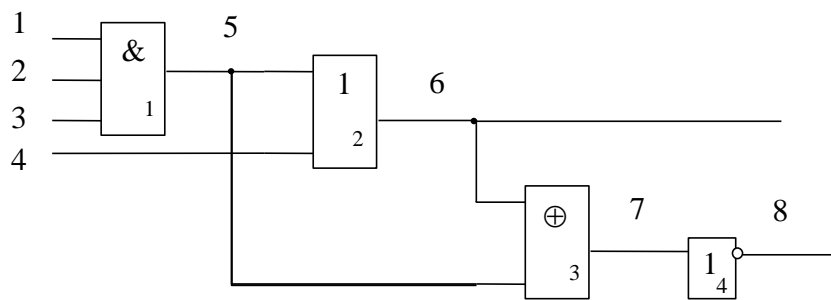


Рисунок 1.2

Значення керованості вузлів не змінилися, тому необхідно перерахувати тільки значення спостережності і тестопридатності вузлів.

$$OY(8-8) = OY(7-8) = OY(6-6) = 1, OY(6-8) = 0.25, OY(6-6) = 1.$$

Для випадку розгалуження виходу розраховують $OY(6-(6,8))$ за формулою

$$OY(X-(Pвих1, Pвих2)) = 1 - ([1 - OY(X-Pвих1)][1 - OY(X-Pвих2)]), \quad (1.7)$$

де $Pвих$ – первинні виходи схеми.

$$OY(6-(6,8)) = 1 - (1 - OY(6-6))(1 - OY(6-8)) = 1 - (1 - 1)(1 - 0,25) = 1,$$

$$OY(5-6,8) = KOY(2)OY(6-6)CY(4) = 0.5,$$

$$OY(5-8) = 0.3125 \text{ (шлях 5-7-8),}$$

$$OY(5-(6,8)) = 1-(1-0,5)(1-0,3125) = 0.65625, OY(4-(6,8)) = KOY(2)OY(6-(6,8))CY(5) = 0.125,$$

$$OY(1-(6,8)) = OY(2-(6,8)) = OY(3-(6,8)) = OY(5-(6,8)) KOY(1) (CY(2)+CY(3))/2 = 0.164.$$

$$TY(1) = 0.164,$$

$$TY(2) = 0.164,$$

$$TY(3) = 0.164,$$

$$TY(4) = 0.125,$$

$$TY(5) = 0.164,$$

$$TY(6) = 0.31342,$$

$$TY(7) = 0.281,$$

$$TY(8) = 0.281.$$

Визначаємо CY^* , OY^* , TY^* . Значення показників після модифікації зазначено в таблиці 1.2.

Таблиця 1.2 – Значення показників після модифікації

Номер вузла	CY^*	OY^*	TY^*
1	1	0.797	0.797
2	1	0.797	0.797
3	1	0.797	0.797
4	1	0.771	0.771
5	0.841	0.9497	0.799
6	0.865	1	0.865
7	0.853	1	0.853
8	0.853	1	0.853

$$TY^* \text{ схеми} = 0.814.$$

2 Визначити керованість, спостережність і тестопридатність вузлів схеми, зображеної на рисунку 1.3. Збільшити тестопридатність вузла з найнижчим її значенням.

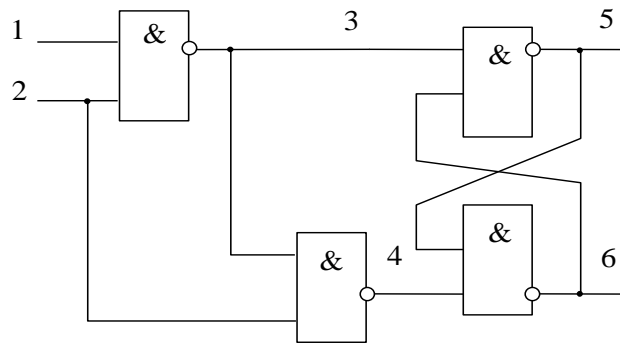


Рисунок 1.3 – DV-триггер

$$K_{CY} = 0.5, \quad K_{OY} = 0.5.$$

$$C_Y(1) = C_Y(2) = 1.$$

$$C_Y(3) = K_{CY} * (C_Y(1) + C_Y(2)) / 2 = 0.5$$

$$C_Y(4) = K_{CY} * (C_Y(2) + C_Y(3)) / 2 = 0.375.$$

$$C_Y(5) = K_{CY} * (C_Y(3) + C_Y(6)) / 2 = 0.158.$$

$$C_Y(6) = K_{CY} * (C_Y(4) + C_Y(5)) / 2 = 0.133.$$

$$O_Y(5-5) = O_Y(6-6) = 1.$$

$$O_Y(5-5,6) = O_Y(6-6,5) = 1.$$

$$O_Y(4-6) = K_{OY} * O_Y(6-6) * C_Y(5) = 0.079.$$

$$O_Y(3-5) = K_{OY} * O_Y(5-5) * C_Y(6) = 0.0665 \text{ (шлях 3-5).}$$

$$O_Y(3-6) = K_{OY} * O_Y(4-6) * C_Y(2) = 0.0395 \text{ (шлях 3-4-6).}$$

$$O_Y(3-(5,6)) = 1 - (1 - O_Y(3-5)) * (1 - O_Y(3-6)) = 1 - (1 - 0.0665) * (1 - 0.0395) = 0.103.$$

$$O_Y(2-5) = K_{OY} * O_Y(3-(5,6)) * C_Y(1) = 0.0515 \text{ (шлях 2-3-5).}$$

$$O_Y(2-6) = K_{OY} * O_Y(4-6) * C_Y(3) = 0.01975 \text{ (шлях 2-4-6).}$$

$$O_Y(2-(5,6)) = 1 - (1 - O_Y(2-5)) * (1 - O_Y(2-6)) = 1 - (1 - 0.0515) * (1 - 0.01975) = 0.0704.$$

$$O_Y(1-(5,6)) = K_{OY} * O_Y(3-(5,6)) * C_Y(2) = 0.0515 \text{ (шлях 1-3-5).}$$

$$T_Y(1) = 0.0515, \quad T_Y(2) = 0.0704, \quad T_Y(3) = 0.0515, \quad T_Y(4) = 0.03,$$

$$T_Y(5) = 0.158, \quad T_Y(6) = 0.133.$$

Значення показників після модифікації зазначено в таблиці 1.3.

Таблиця 1.3 – Значення показників після модифікації

Номер вузла	СУ*	ОУ*	ТУ*
1	1	0.69	0.69
2	1	0.718	0.718
3	0.917	0.752	0.69
4	0.885	0.728	0.645
5	0.794	1	0.794
6	0.777	1	0.777

ТУ* схеми = 0.719.

Контрольні запитання

- 1 Що таке коефіцієнт передавання керованості? Як його визначають?
- 2 Що таке коефіцієнт передавання спостережності? Як його визначають?
- 3 Що таке одновимірні неполяризовані активізуючі і блокуючі D-куби? Указати такі куби для логічних елементів 2хІ, 2хАБО, 3хІ-НЕ, 3хАБО-НЕ.
- 4 Як визначають керованість вузлів схеми?
- 5 Вказати, як визначають спостережність вузлів схеми у випадках:
а) розгалуження виходу; б) розгалужень, що сходяться; в) схеми з колами зворотного зв'язку.
- 6 Як визначають тестопридатність схеми в цілому?
- 7 Які існують основні способи покращення тестопридатності цифрових логічних схем?

Варіанти завдань наведено в таблиці 1.4

Таблиця 1.4

Номер з/п	Комбінаційна схема або послідовна схема	Номер з/п	Комбінаційна схема або послідовна схема
1		6	
2		7	
3		8	
4		9	
5		10	

Продовження таблиці 1.4

Номер з/п	Комбінаційна схема або послідовна схема	Номер з/п	Комбінаційна схема або послідовна схема
11		15	
12		16	
13		17	
14		18	

2 ПРОЄКТУВАННЯ ТЕСТОПРИДАТНИХ ЦИФРОВИХ АВТОМАТІВ МЕТОДОМ СКАНОВАНОГО ШЛЯХУ

Мета: набути навичок у забезпеченні тестопридатності цифрових автоматів методами шляхів, сканованих ScanPath (SP).

Завдання

Набути навичок у забезпеченні тестопридатності цифрових автоматів методом шляху, сканованого (Scan Path).

Приклад виконання завдання

Побудувати тестопридатну схему автомата Мура, описаного граф-схемою алгоритму ГСА (рисунок 2.1) і методом сканованого шляху (SP). Для побудови схеми використовувати елементну базу I, АБО, НЕ. Для зменшення розмірності задачі одержання тестової мікропрограми використовувати в автоматі синхронні D-тригери без інверсних виходів; реалізацію функцій збудження тригерів виконати без переддешифрації; функції виходів опустити.

Розмітимо ГСА, як на рисунку 2.1.

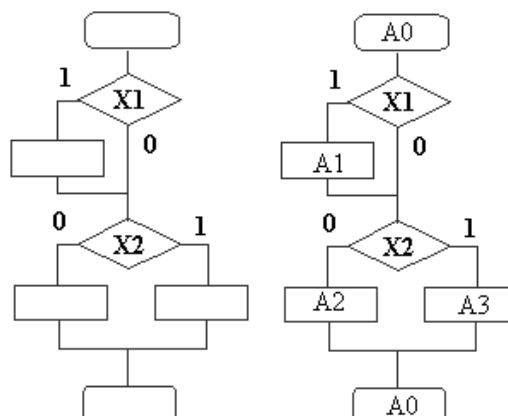


Рисунок 2.1 – ГСА

У першому наближенні автомат повинен мати два інформаційних входи x_1 , x_2 , вхід синхронізації C , вхід початкової установки R , виходи управління Y і дві внутрішні лінії зворотного зв'язку Q_1 і Q_2 . Для спрощення задачі відповідно до вихідних вимог формування керуючих сигналів розглядати не будемо. Розглянемо автомат Мура (рисунок 2.2).

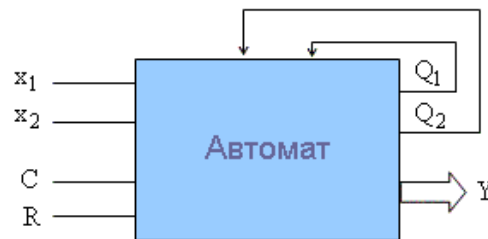


Рисунок 2.2 – Автомат Мура

Для синтезу автомата необхідно побудувати пряму (або зворотну) структурну таблицю переходів станів. Стовпець $Y(a_m, a_s)$ опускаємо (таблиця 2.1).

Таблиця 2.1 – Пряма структурна таблиця переходів

№	t		t+1		$X(a_m, a_s)$	$F(a_m, a_s)$	
	a_m	$k(a_m)$	a_s	$k(a_s)$		D_1	D_2
1	a_0	00	a_2	10	$\bar{x}_1 \bar{x}_2$	1	0
2			a_3	11	$\bar{x}_1 x_2$	1	1
3			a_1	01	x_1	0	1
4	a_1	01	a_2	10	\bar{x}_2	1	0
5			a_3	11	x_2	1	1
6	a_2	10	a_0	00	–	0	0
7	a_3	11	a_0	00	–	0	0

Функції збудження тригерів, отримані зі структурної таблиці переходів, мають вирази

$$D_1 = a_0 \bar{x}_1 \bar{x}_2 \vee a_0 \bar{x}_1 x_2 \vee a_1 \bar{x}_2 \vee a_1 x_2 = a_0 \bar{x}_1 \vee a_1, \quad (2.1)$$

$$D_2 = a_0 \bar{x}_1 x_2 \vee a_0 x_1 \vee a_1 x_2 = a_0 (\bar{x}_1 x_2 \vee x_1) \vee a_1 x_2 = a_0 (x_2 \vee x_1) \vee a_1 x_2. \quad (2.2)$$

Для реалізації автомата без переддешифрації стану автомата a_i необхідно подати його у вигляді кон'юнкції Q_1 і Q_2 , обумовленої кодом $k(a_i)$. Так, для $k(a_2) = 10$ $Q_1 = 1$ і $Q_2 = 0$. А оскільки всі рівняння у процесі синтезу записують у вигляді диз'юнктивної нормальної форми (ДНФ), вираз для стану a_2 буде мати подібний вигляд. Тепер функції збудження тригерів будуть мати вигляд

$$D_1 = \bar{Q}_1 \bar{Q}_2 \bar{x}_1 \vee \bar{Q}_1 Q_2; \quad (2.3)$$

$$D_2 = \bar{Q}_1 \bar{Q}_2 (x_2 \vee x_1) \vee \bar{Q}_1 Q_2 x_2. \quad (2.4)$$

Такий спосіб рекомендований для лабораторної роботи і буде використаний далі.

Реалізація автомата без переддешифрації подана на рисунку 2.3, де КС – комбінаційна частина схеми. З міркувань компактності і наочності схеми інвертори на входах вентилів замінені інверсними входами.

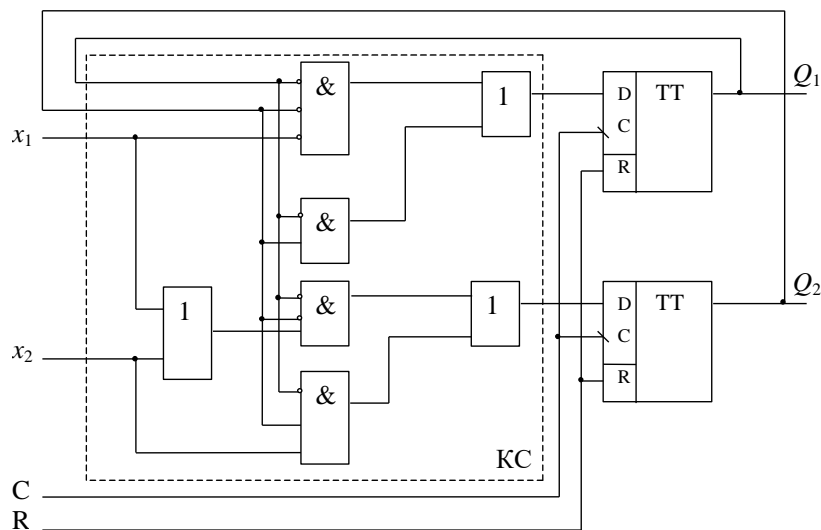


Рисунок 2.3 – Реалізація тестопридатного автомата без переддешифрації

Реалізація тестопридатного автомата методом сканованого шляху подана на рисунку 2.4, де:

- MX – мультиплексор, необхідний для поділу режимів нормального функціонування і тестованості;
- за $V = 0$ вибирають режим нормального функціонування (F) – мультиплексори з'єднують виходи КС із входами тригерів;
- за $V = 1$ вибирають режим сканування (SP) – елементи пам'яті реконфігуровані в один зсувний регістр із послідовним введенням – виведенням даних за $V = 1$;
- X_s – вхід сканування даних, на нього дані надходять послідовно, біт за бітом;
- Q_1 – вихід сканування даних, з нього послідовно зчитують дані, що зрушуються.

Отже, у режимі SP елементи пам'яті можна досить просто установити в будь-який заданий стан, подаючи послідовність сигналів на X_s , тактуючи зсувний регістр за допомогою тактового імпульсу С.

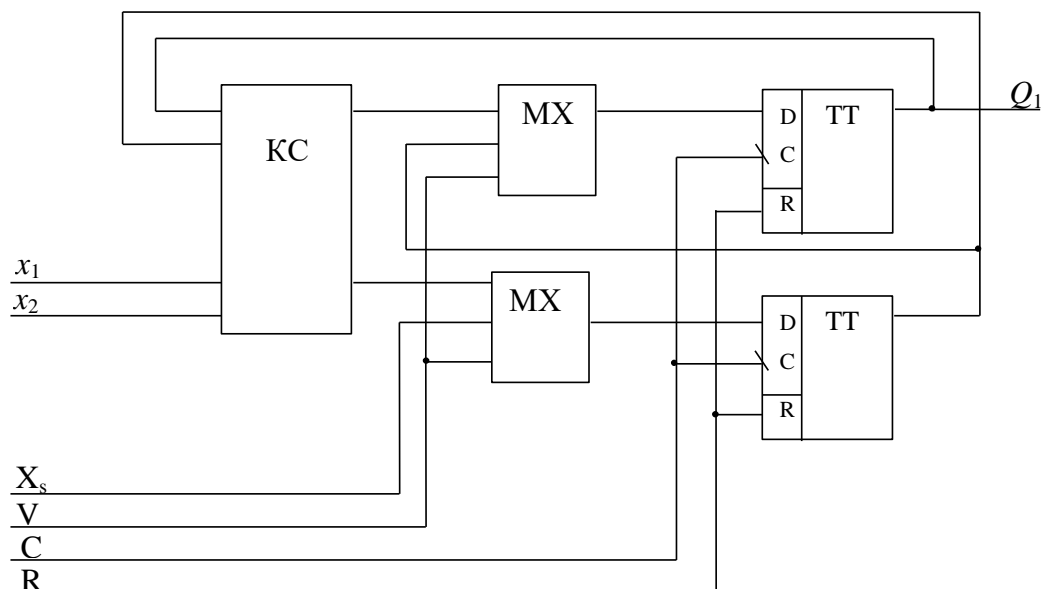


Рисунок 2.4 – Реалізація тестопридатного автомата методом сканованого шляху

Тепер розглянемо побудову тестової мікропрограми.

Для початку необхідно побудувати якомога повний мінімальний тест, що перевіряє одиночну константну несправність (ОКН), для комбінаційної схеми (КС) з огляду на те, що зовнішні входи X і внутрішні змінні Q будуть безпосередньо керовані, а виходи КС безпосередньо спостережувані, тобто КС буде мати первинні входи XQ (x1, x2, Q1, Q2). Побудова мінімального тесту здійснюється або D-алгоритмом, або K-алгоритмом, або за таблицею несправностей.

У таблиці 2.2 наведена тестова мікропрограма. Е – передній фронт (0/1). Сірим кольором виділені фрагменти повного мінімального тесту для КС.

Тестова мікропрограма – послідовність мікрокоманд управління режимом тестованості й аналізу результатів тестованості.

Таблиця 2.2

	Коментар до режиму	Тестова мікропрограма					Вихід тригера Q1 Q2
		R	C	V	Xs	x1x2	
1	2	3	4	5	6	7	8
0	Початкова установка тригерів	1	E	x	x	x x	00
1	SP Тест А (рухома 1)	0	E	1	1	x x	10
		0	E	1	0	x x	01
		0	E	1	0	x x	00
2	SP Тест А (рухомий 0)	0	E	1	1	x x	10
		0	E	1	1	x x	11
		0	E	1	0	x x	01
		0	E	1	1	x x	10
		0	E	1	1	x x	11

Продовження таблиці 2.2

1	2	3	4	5	6	7	8
3	SP Тест Б (рухомі пари 001100)	0	E	1	0	x x	01
		0	E	1	0	x x	00
		0	E	1	1	x x	10
		0	E	1	1	x x	11
		0	E	1	0	x x	10
		0	E	1	0	x x	00
4	SP Зсув останніх бітів тесту Б і запис частини Q ($Q_2Q_1 = 11$) тестового вектора 1	0	E	1	1	x x	10
		0	E	1	1	x x	11
5	F Подання вектора 1 ($x_1x_2Q_1Q_2 = 1011$) на входи КС	0	E	1	x	1 0	00
6	SP Зсув відклику на вектор 1 (Q_1Q_2) і запис частини Q ($Q_2Q_1 = 00$) тестового вектора 2	0	E	1	0	x x	00
		0	E	1	0	x x	00
7	F Подання вектора 2 ($x_1x_2Q_1Q_2 = 1100$) на входи КС	0	E	1	x	1 1	10
8	SP Зсув відклику на вектор 2 (Q_1Q_2) і запис частини Q ($Q_2Q_1 = 10$) тестового вектора 3	0	E	1	1	x x	10
		0	E	1	0	x x	01
9	F Подання вектора 3 ($x_1x_2Q_1Q_2 = 1010$) на входи КС	0	E	0	x	1 0	00
10	SP Зсув відклику на вектор 3 (Q_1Q_2) і запис частини Q ($Q_2Q_1 = 11$) тестового вектора 4	0	E	1	1	x x	10
		0	E	1	1	x x	11
11	F Подання вектора 4 ($x_1x_2Q_1Q_2 = 0111$) на входи КС	0	E	0	x	1 0	00
12	SP Зсув відклику на вектор 4	0	E	1	x	x x	x 0

Контрольні запитання

1 Поясніть принцип забезпечення тестопридатності цифрових автоматів методом сканованого шляху.

2 Які тести використовують для тестованості комбінаційної і послідовних схем цифрового автомата за використання методу сканованого шляху?

3 Що таке тестопридатне проектування (ТПР)?

4 Чим структурні методи ТПР відрізняються від спеціалізованих?

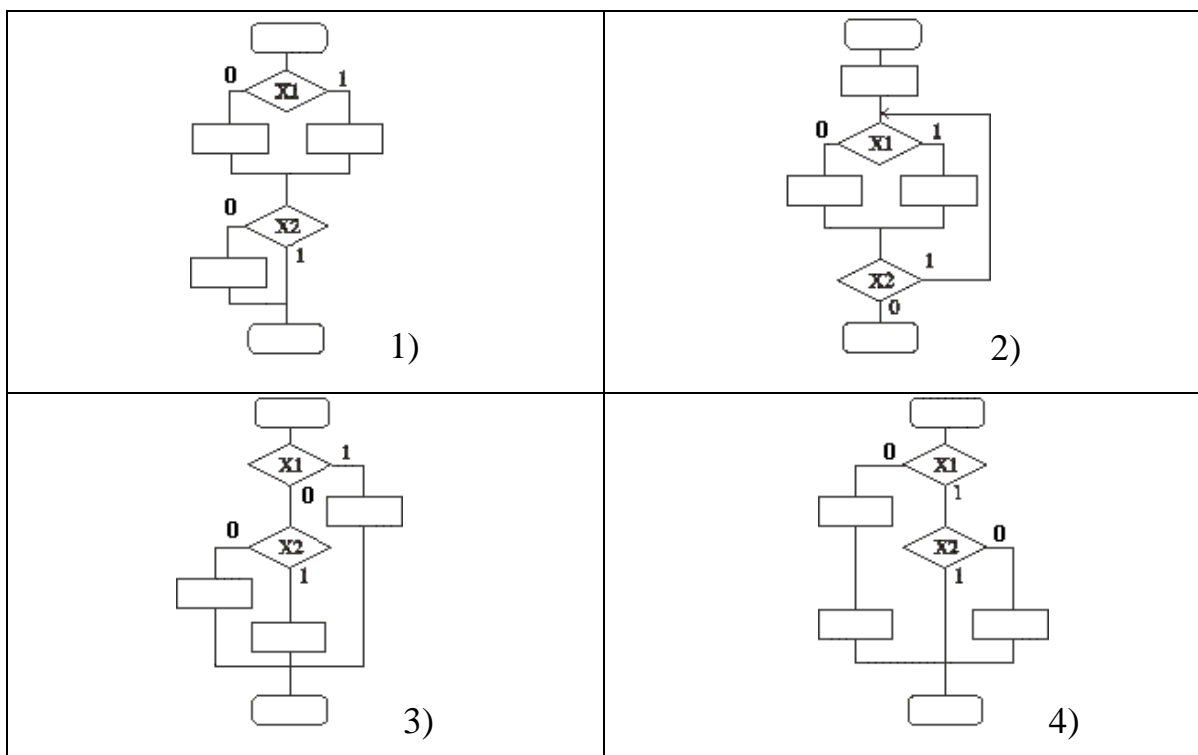
5 У чому полягає ідея сканованого шляху?

6 Перерахуйте недоліки і переваги методів сканування.

7 Що таке тестова мікропрограма?

Варіанти завдань для самостійного виконання наведено в таблиці 2.3.

Таблиця 2.3



Продовження таблиці 2.3

<p>5)</p>	<p>6)</p>
<p>7)</p>	<p>8)</p>
<p>9)</p>	<p>10)</p>
<p>11)</p>	<p>12)</p>
<p>13)</p>	<p>14)</p>

3 СИГНАТУРНІ АНАЛІЗАТОРИ В СИСТЕМАХ ВБУДОВАНОГО САМОТЕСТУВАННЯ

Мета: вивчити різновиди і особливості сигнатурних аналізаторів, що найчастіше використовують у системах вбудованого самотестування.

3.1 Сигнатурні аналізатори з внутрішніми елементами XOR у системах вбудованого самотестування

Завдання

Набути навичок будування сигнатурних аналізаторів із внутрішніми елементами XOR у системах вбудованого самотестування.

Приклад виконання завдань

1 Дано характеристичний поліном $P^*(x) = 1 + x^3 + x^5 + x^7 + x^9$, що описує ЗРЛЗЗ із внутрішніми елементами XOR. Зобразити схему цього ЗРЛЗЗ у вигляді сигнатурного аналізатора і визначити сигнатуру вхідної послідовності (10111001011100000→) за умови, що тригери ЗРЛЗЗ скинуті в нуль.

Розв'язання

$P^*(x) = 1 + x^3 + x^5 + x^7 + x^9$ – зворотний характеристичний поліном (із зірочкою), йому відповідає прямий характеристичний поліном (без зірочки) $P(x) = x^9 + x^6 + x^4 + x^2 + 1$. Обидва поліноми описують зворотні зв'язки в ЗРЛЗЗ, однак зворотний характеристичний поліном зручніше використовувати для відновлення структури ЗРЛЗЗ – показник степеня доданків вказує на номер тригера, за яким впливає елемент XOR.

Схема сигнатурного аналізатора зображена на рисунку 3.1.

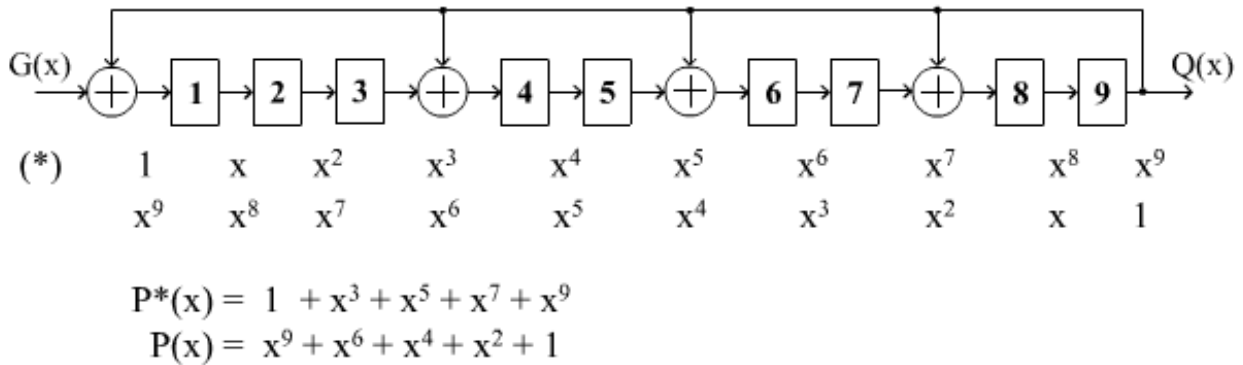


Рисунок 3.1 - Схема сигнатурного аналізатора

Вхідну (ущільнювану) послідовність 10111001011100000 описують утворюючим поліномом $G(x) = 1 + x^2 + x^3 + x^4 + x^7 + x^9 + x^{10} + x^{11}$. Стрілка вказує напрямок потоку інформації. Перший ущільнюваний біт (біля стрілки) відповідає максимальному показнику степеня, а останній біт – мінімальному показнику (0) у $G(x)$. Степінь записують у багаточлен, якщо йому відповідає одиниця у двійковій послідовності.

Сигнатура – це вміст ЗРЛЗЗ після ущільнення всіх бітів вхідної послідовності. З математичної точки зору, для аналізатора з внутрішніми елементами XOR сигнатура – це залишок (частка(частное)/остача(остаток)?математика) від розподілу утворюючого полінома на зворотний характеристичний поліном. Тому для визначення сигнатури досить зробити поліноміальний розподіл $G(x)$ на $P^*(x)$.

$$x^{11} + x^{10} + x^9 + x^7 + x^4 + x^3 + x^2 + 1 \mid \frac{x^9 + x^7 + x^5 + x^3 + 1}{x^2 + x};$$

$$\frac{x^{11} + x^9 + x^7 + x^5 + x^2}{x^{10} + x^5 + x^4 + x^3 + 1};$$

$$\frac{x^{10} + x^8 + x^6 + x^4 + x}{x^8 + x^6 + x^5 + x^3 + x + 1}.$$

Залишок $R(x) = 1 + x + x^3 + x^5 + x^6 + x^8$ відповідає сигнатурі 110101101.

Відповідність розрядів регістра і розмірів полінома наведена в таблиці 3.1.

Таблиця 3.1

Номер тригера	1	2	3	4	5	6	7	8	9
Степінь $R(x)$	0	1	2	3	4	5	6	7	8
Сигнатура	1	1	0	1	0	1	1	0	1

3.2 Сигнатурні аналізатори з зовнішніми елементами XOR у системах вбудованого самотестування

Завдання

Набути навичок будування сигнатурних аналізаторів із зовнішніми елементами XOR у системах вбудованого самотестування.

Дано характеристичний поліном $P^*(x) = x^4 + x + 1$, що описує ЗРЛЗЗ із зовнішніми елементами XOR. Зобразити схему цього ЗРЛЗЗ у вигляді сигнатурного аналізатора і визначити сигнатуру вхідної послідовності 111110 за умови, що тригери ЗРЛЗЗ скинуті в нуль.

Розв'язання

$P^*(x) = x^4 + x + 1$ – зворотний характеристичний поліном (із зірочкою), йому відповідає прямий характеристичний поліном (без зірочки) $P(x) = 1 + x^3 + x^4$.

Обидва поліноми описують зворотні зв'язки в ЗРЛЗЗ, однак прямий характеристичний поліном зручніше використовувати для відновлення структури ЗРЛЗЗ – показник степеня доданків вказує на номер тригера, із виходу якого йде зворотний зв'язок на зовнішній елемент XOR. Схема сигнатурного аналізатора зображена на рисунку 3.2.

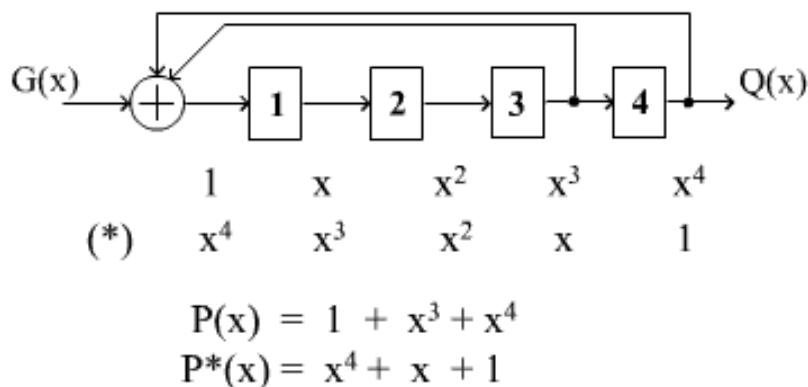


Рисунок 3.2

Вхідну (ущільнювану) послідовність 111110 описують утворюючим поліномом $G(x) = 1 + x + x^2 + x^3 + x^4$. Стрілка вказує напрямок потоку інформації. Перший ущільнюваний біт (біля стрілки) відповідає максимальному показнику степеня, а останній біт – мінімальному показнику (0) у $G(x)$. Степінь записують у багаточлен, якщо йому відповідає одиниця у двійковій послідовності.

Сигнатура – це вміст ЗРЛЗЗ після ущільнення всіх бітів вхідної послідовності. Цей тип сигнатурного аналізатора також виконує поліноміальний розподіл утворюючого полінома $G(x)$ на зворотний характеристичний поліном $P^*(x)$, при цьому отримують коректну частку, однак вмістом регістра після розподілу буде не залишок $R(x)$. Залишок $R(x)$ може бути отриманий тільки в окремих випадках. Тому для визначення сигнатури необхідно зробити моделювання роботи аналізатора.

Результати моделювання ущільнення вхідної послідовності наведено в таблиці 3.2.

Таблиця 3.2

Час	Вхідна послідовність	Зміст регістра	Вихідна послідовність
	G(x) →	1 2 3 4	Q(x) →
0	1 1 1 1 1 0	0 0 0 0	
1	1 1 1 1 1	0 0 0 0	
2	1 1 1 1	1 0 0 0	
3	1 1 1	1 1 0 0	
4	1 1	1 1 1 0	
5	1	0 1 1 1	0
6		0 1 1 1	10
		$R(x) = x + x^2 + x^3$	$Q(x) = 1$

Сигнатура – зміст регістра на шостому такті – 0111.

Виконаємо поліноміальний розподіл у стовпчик:

$$\begin{array}{r}
 x^4 + x^3 + x^2 + x + 1 \mid \underline{x^4 + x + 1} \\
 \underline{x^4 + x + 1} \qquad \qquad \qquad 1 \\
 x^3 + x^2
 \end{array}$$

За наведеним розподілом виділимо:

- 1) частка $Q(x) = 1$;
- 2) залишок $R(x) = x^3 + x^2$.

Якщо $R(x) \neq R^{\wedge}(x)$, то в такому випадку сигнатура є залишком (таблиця 3.2).

Контрольні запитання

1 За допомогою яких пристроїв реалізовано генератори тестових дій у системах вбудованого самотестування?

2 За допомогою чого можна отримати псевдовипадковий тест:

- 1) лічильника;
- 2) зсувного регістра;
- 3) зсувного регістра за лінійними зворотними зв'язками?

3 Чому дорівнює максимальний період псевдовипадкової послідовності, що генерується n -розрядним регістром:

- 1) $2n-1$;
- 2) $2n+1$;
- 3) $2n$?

4 За допомогою яких пристроїв реалізують сигнатурні аналізатори в системах вбудованого самотестування (ВСТ)?

5 Що описують утворюючим поліномом:

- 1) зворотні зв'язки зсувного регістра з лінійним зворотним зв'язком (ЗРЛЗЗ);
- 2) ущільнювану послідовність;
- 3) вихідну послідовність?

6 Для якого ЗРЛЗЗ справедливо, що сигнатура – залишок від ділення утворюючого полінома на зворотний характеристичний із зовнішніми або внутрішніми елементами XOR.

Варіанти завдань для самостійного виконання наведено в таблиці 3.3.

Таблиця 3.3

Варіант	Характеристичний поліном	Вхідна послідовність
із внутрішніми елементами XOR		
1	$P(x) = x^9 + x^7 + x^3 + x + 1$	00010111011101110 →
2	$P(x) = x^8 + x^6 + x^2 + 1$	00110011010110101 →
3	$P(x) = x^8 + x^5 + x^4 + x + 1$	11001100001100110 →
4	$P(x) = x^9 + x^5 + x^3 + x^2 + 1$	10110110000001000 →
5	$P(x) = x^9 + x^3 + x^2 + 1$	11100011001010111 →
6	$P(x) = x^{10} + x^8 + x^4 + 1$	00001100110101010 →
7	$P(x) = x^9 + x^5 + x^4 + 1$	01100110011011000 →
8	$P(x) = x^{10} + x^6 + x^3 + 1$	01000111100101010 →
9	$P(x) = x^9 + x^8 + x^5 + x^2 + 1$	00001110010100011 →
10	$P(x) = x^{10} + x^7 + x^4 + x^3 + 1$	00000011111010101 →
11	$P(x) = x^8 + x^7 + x^5 + x^2 + 1$	00100101110001100 →
12	$P(x) = x^9 + x^7 + x^6 + x^3 + 1$	10101011111100001 →
13	$P(x) = x^9 + x^6 + x^4 + x^2 + 1$	10111001011100010 →
із зовнішніми елементами XOR		
1	$P^*(x) = x^6 + x^2 + x + 1$	111011001
2	$P^*(x) = x^8 + x^3 + 1$	111100001
3	$P^*(x) = x^8 + x^4 + 1$	100110110
4	$P^*(x) = x^6 + x^3 + x^2 + 1$	1101100011
5	$P^*(x) = x^6 + x^4 + x + 1$	101101001
6	$P^*(x) = x^7 + x^3 + x^2 + 1$	1100011100
7	$P^*(x) = x^6 + x^5 + x^4 + 1$	011001101
8	$P^*(x) = x^7 + x^6 + x^3 + 1$	001111001
9	$P^*(x) = x^8 + x^7 + 1$	001110010
10	$P^*(x) = x^6 + x^3 + 1$	00111110101
11	$P^*(x) = x^8 + x^2 + 1$	1110001100
12	$P^*(x) = x^8 + x + 1$	011001100
13	$P^*(x) = x^6 + x^4 + x^2 + 1$	001011101

4 ЗАБЕЗПЕЧЕННЯ С-ТЕСТОВАНОСТІ І ПОБУДОВА ТЕСТУ ДЛЯ СИСТОЛІЧНИХ МАТРИЦЬ КОМБІНАЦІЙНИХ ЕЛЕМЕНТІВ

Мета заняття: набути навичок у забезпеченні С-тестованості ортогональних одно- і двовимірних систолічних матриць комбінаційних елементів (СМКЕ), функція комірки яких описана таблицею переходів станів, а також у побудові тестового забезпечення для СМКЕ.

Приклади виконання завдань

1 Дано одновимірну односпрямовану СМКЕ, що складається з 11 комірок, з функцією комірки, заданої базовою таблицею переходів S_{b1} . Забезпечити С-тестованості СМКЕ, модифікуючи базову таблицю переходів $(1,7) \rightarrow 3$. Побудувати тест для зазначеного переходу. Оцінити час тестованості матриці за базовою таблицею переходів S_{b1} (таблиця 4.1).

Таблиця 4.1

		Y							
		0	1	2	3	4	5	6	7
X	0	0	1	3	2	1	1	2	3
	1	2	0	2	0	0	1	1	3
	2	3	1	1	3	1	0	1	2
	3	3	3	0	0	2	2	0	0

Розв'язання

Для забезпечення С-тестованості одновимірної односпрямованої СМКЕ модифікуємо функцію комірки за такими правилами:

$$1) f^*(x_i, y_{n+1}) = x_k, \quad 1 \leq i \leq m+1, 1 \leq k \leq m+1;$$

2) $f(x_{m+1}, y_{n+2}) = x_r, \quad 1 \leq r \leq m$ (вибираємо тільки одне r з m можливих);

3) $f(x_s, y_{n+2}) = x_{m+1}, \quad 1 \leq s \leq m;$

4) $f^*(x_i, y_{n+3}) = x_k, \quad 1 \leq i \leq m, 1 \leq k \leq m,$ де m – кількість x -станів у S_{b1} ;

5) для інших x та y в S_{m1} $f(x,y) = x$.

Модифікована таблиця переходів (таблиця 4.2) S_{m1} побудована на основі базової таблиці переходів S_{b1} і правил заповнення (1-5).

Таблиця 4.2

		y_{n+}															
		1	2	3	4	5	6	7	8								
		0	1	2	3	4	5	6	7								
		8	9	10	11	12	13	14	15								
x_1	0	0	1	3	2	1	1	2	3	1	4	1	0	0	0	0	0
x_2	1	2	0	2	0	0	1	1	3	2	4	2	1	1	1	1	1
x_3	2	3	1	1	3	1	0	1	2	3	4	3	2	2	2	2	2
x_4	3	3	3	0	0	2	2	0	0	4	4	0	3	3	3	3	3
x_5	4	4	4	4	4	4	4	4	4	0	0	3	4	4	4	4	4
x_6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
x_7	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
x_8	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

Тестованості переходу $(1,7) \rightarrow 3$ для заштрихованих комірок показано на рисунку 4.1.

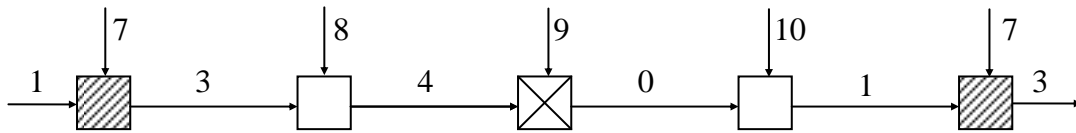


Рисунок 4.1 – Тестованості переходу $(1,7) \rightarrow 3$
для заштрихованих комірок

Отже, тестова послідовність і вектор моделювання для переходу $(1,7) \rightarrow 3$ мають такий вигляд:

7 8 9 10 – тестова послідовність (у x-напрямку подавана на у-входи);
1 3 4 0 – вектор моделювання (у x-напрямку).

Жирним шрифтом виділені значення, подавані на входи матриці. Вони відповідають тестовому вектору для першої (заштрихованої) комірки. Щоб одержати тестові вектори для інших комірок послідовності, що регенерує, необхідно тестову послідовність і вектор моделювання зрушувати одночасно циклічно на один розряд вправо.

7 8 9 10 ← тестовий вектор для першої (заштрихованої) комірки;
1 3 4 0 ← вектор моделювання;

10 **7 8 9** ← тестовий вектор для другої комірки, отриманий циклічним зрушенням на один розряд вправо всіх розрядів тесту і вектора моделювання для першої комірки;

0 **1 3 4**;

9 10 **7 8** ← тестовий вектор для третьої комірки;

4 0 **1 3**;

8 9 10 **7** ← тестовий вектор для четвертої комірки;

3 4 0 **1**.

Отримані тестові вектори застосовані до кожної четвірки наступної комірки уздовж усієї матриці, наприклад для тестованості першої, п'ятої, дев'ятої комірки в СМКЕ, що містить 11 комірок. Вхідний (тестовий) вектор буде виглядати так:

7 8 9 10 7 8 9 10 7 8 9 (в у-напрямку);

1 (в х-напрямку).

Для другої, шостої, десятої:

10 7 8 9 10 7 8 9 10 7 8;

0.

Для третьої, сьомої, одинадцятої:

9 10 7 8 9 10 7 8 9 10 7;

4.

Для четвертої і восьмої:

8 9 10 7 8 9 10 7 8 9 10;

3.

Час тестованості (кількість тестових векторів) розраховують за формулою

$$T = 4mn(2m+1), \quad (4.1)$$

де m – кількість х-станів;

n – кількість у-станів.

$m = 4, n = 8$, тому $T = 1152$.

2 Дано двовимірну односпрямовану ортогонально зв'язану СМКЕ із функцією комірки, заданою базовою таблицею переходів S_{b2} (таблиця 4.3). Забезпечити С-тестованість СМКЕ, модифікуючи базову таблицю переходів. Побудувати тест для зазначеного переходу $(2,5) \rightarrow (3,5)$. Оцінити час тестованості матриці. Побудувати тест для переходу.

Таблиця 4.3 – Базова таблиця переходів S_{b_2}

		Y			
		0	1	2	3
X	0	(1,2)	(2,3)	(3,1)	(3,0)
	1	(2,3)	(2,1)	(1,3)	(0,0)
	2	(1,2)	(0,3)	(0,0)	(1,0)
	3	(3,1)	(1,2)	(0,2)	(0,1)

Розв'язання

Модифікована таблиця переходів $S_{m_{2x}}$ (таблиця 4.4) побудована на основі базової таблиці переходів S_{b_2} (таблиця 4.3) і правил заповнення (1–5).

Для у-напрямку використовують ті самі правила заповнення (1 – 5), тільки в них x та y міняють місцями, а замість m записують n , n – кількість у-станів у S_{b_2} . Зверніть увагу на особливість розшарування базової таблиці переходів S_{b_2} (таблиця 4.5) – рядки у-значень з таблиці S_{b_2} відповідають стовпцям у таблиці $S_{m_{2y}}$.

Таблиця 4.4 – Модифікована таблиця переходів $S_{m_{2x}}$ для х-напрямку

						Y _n	Y _n	Y _n	
		0	1	2	3	+2	+3	+4	
x	0	1	2	3	3	0	1	4	1
	1	2	2	1	0	1	2	4	2
	2	1	0	0	1	2	3	4	3
	3	3	1	0	0	3	4	4	0
	4	4	4	4	4	4	0	0	4
	5	5	5	5	5	5	5	5	5
	6	6	6	6	6	6	6	6	6
	7	7	7	7	7	7	7	7	7

Таблиця 4.5 – Модифікована таблиця переходів S_{m2y} для у-напрямку

					x_n	x_n	x_n		
					+2	+3	+4		
		0	1	2	3	4	5	6	7
y_1	0	2	3	2	1	0	1	4	1
y_2	1	3	1	3	2	1	2	4	2
y_3	2	1	3	0	2	2	3	4	3
y_4	3	0	0	0	1	3	4	4	0
y_5	4	4	4	4	4	4	0	0	4
y_6	5	5	5	5	5	5	5	5	5
y_7	6	6	6	6	6	6	6	6	6
y_8	7	7	7	7	7	7	7	7	7

Тестованості переходу $(2,5) \rightarrow (3,5)$ для заштрихованих комірок показано на рисунку 4.2.

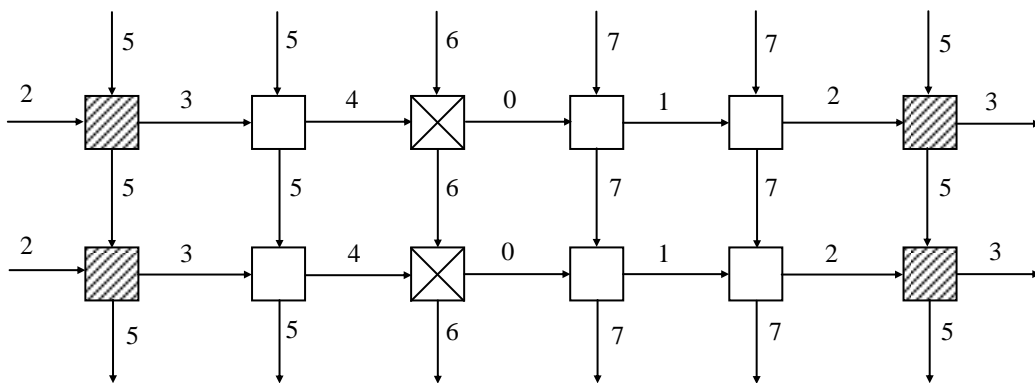


Рисунок 4.2

Отже, тестова послідовність і вектор моделювання для переходу $(2,5) \rightarrow (3,5)$ у х-напрямку має такий вигляд:

5 5 6 7 7 - тестова послідовність;

2 3 4 0 1 – вектор моделювання.

А в у-напрямку:

2 – тестова послідовність; **5** – вектор моделювання.

Жирним шрифтом виділені значення, подавані на входи матриці. Вони відповідають тестовому вектору для заштрихованих комірок. Щоб одержати тестові вектори для інших комірок послідовності, що регенерує, у х-напрямку, необхідно тестову послідовність і вектор моделювання зрушувати одночасно циклічно на один розряд вправо. Потім циклічно зрушують тестову послідовність в у-напрямку на один розряд, і знову провадять всі зрушення в х-напрямку тощо доти, поки не здійсниться останнє зрушення в у-напрямку. Одержувані в такий спосіб вхідні набори застосовують до наступних комірок уздовж усієї матриці. Для розглянутого переходу тестову послідовність в у-напрямку не зрушують, оскільки в цьому напрямку регенерація тестового входу не потрібна.

Час тестованості матриці (кількість тестових векторів) можна оцінити за формулою

$$T = 4mn(2m+1)(2n+1), \quad (4.2)$$

де m – кількість х-станів;

n – кількість у-станів.

$m = 4$, $n = 4$, тому $T = 5184$.

Контрольні запитання

- 1 Які матричні структури називають систолами?
- 2 Що зумовило розвиток систолічних структур? Де вони застосовані?
- 3 Яку функцію називають добре визначеною?

4 Сформулюйте умови забезпечення С-тестованості одновимірних ортогональних СМКЕ.

5 Сформулюйте правила модифікації базової таблиці переходів і побудови тесту для забезпечення С-тестованості одно- і двовимірних СМКЕ.

Варіанти завдань для самостійного виконання

Завдання 4.1. Дано одновимірну односпрямовану СМКЕ, що складається з заданої кількості комірок, з функцією комірки, заданою базовою таблицею переходів. Забезпечити С-тестованість СМКЕ, модифікуючи базову таблицю переходів. Побудувати тест для всіх комірок матриці для заданого переходу. Зобразити фрагмент матриці в рамках однієї регенеруючої послідовності. Оцінити час тестованості матриці. Варіанти завдань наведені в таблиці 4.6.

Базові таблиці переходів S_{b1} та S_{b2} наведені в таблицях 4.7 і 4.8.

Таблиця 4.6

Варіант	Кількість комірок	Базова таблиця переходів	Перехід
1	2	3	4
1	11	S_{b1}	$(0,0) \rightarrow 0$
2	12	S_{b1}	$(0,1) \rightarrow 1$
3	13	S_{b1}	$(0,2) \rightarrow 3$
4	14	S_{b1}	$(0,3) \rightarrow 2$
5	15	S_{b1}	$(0,4) \rightarrow 1$
6	16	S_{b1}	$(0,5) \rightarrow 1$
7	11	S_{b1}	$(0,6) \rightarrow 2$
8	12	S_{b1}	$(0,7) \rightarrow 3$
9	13	S_{b1}	$(1,0) \rightarrow 2$
10	14	S_{b1}	$(1,1) \rightarrow 0$
11	15	S_{b1}	$(1,2) \rightarrow 2$

Продовження таблиці 4.6

1	2	3	4
12	16	S_{b1}	(1,3)→0
13	11	S_{b1}	(1,4)→1
14	12	S_{b1}	(1,5)→0
15	13	S_{b1}	(1,6)→1
16	14	S_{b1}	(1,7)→3
17	15	S_{b1}	(2,0)→3
18	16	S_{b1}	(2,1)→1
19	11	S_{b1}	(2,2)→1
20	12	S_{b1}	(2,3)→3

Таблиця 4.7 – Базова таблиця переходів S_{b1}

		Y							
		0	1	2	3	4	5	6	7
X	0	0	1	3	2	1	1	2	3
	1	2	0	2	0	0	1	1	3
	2	3	1	1	3	1	0	1	2
	3	3	3	0	0	2	2	0	0

Таблиця 4.8 – Базова таблиця переходів S_{b2}

		Y							
		0	1	2	3	4	5	6	7
X	0	3	3	2	0	2	3	1	2
	1	3	1	1	3	2	0	2	0
	2	2	0	0	2	0	2	0	3
	3	1	0	3	2	1	0	1	1

Завдання 4.2. Дано одновимірну односпрямовану СМКЕ, що складається з 15 комірок, з функцією комірки, заданою базовою таблицею переходів S_{b3} . Забезпечити С-тестованість СМКЕ, модифікуючи базову таблицю переходів. Побудувати тест для переходу $(2,4) \rightarrow 3$. Оцінити час тестованості матриці.

Завдання 4.3. Дано двовимірну односпрямовану ортогонально зв'язану СМКЕ з функцією комірки, заданою базовою таблицею переходів S_{b4} . Забезпечити С-тестованість СМКЕ, модифікуючи базову таблицю переходів. Побудувати тест для переходу $(3,3) \rightarrow (2,1)$. Оцінити час тестованості матриці. Варіанти завдань для самостійного виконання наведено в таблиці 4.9.

Таблиця 4.9

Варіант	Перехід	Варіант	Перехід
1	$(1,6) \rightarrow 1$	11	$(0,0) \rightarrow (1,2)$
2	$(5,6) \rightarrow 5$	12	$(1,2) \rightarrow (1,3)$
3	$(3,2) \rightarrow 0$	13	$(3,2) \rightarrow (0,2)$
4	$(7,10) \rightarrow 7$	14	$(1,3) \rightarrow (0,0)$
5	$(3,8) \rightarrow 4$	15	$(1,0) \rightarrow (2,3)$
6	$(4,9) \rightarrow 0$	16	$(4,5) \rightarrow (0,5)$
7	$(3,10) \rightarrow 0$	17	$(3,6) \rightarrow (4,6)$
8	$(6,6) \rightarrow 6$	18	$(6,4) \rightarrow (6,0)$
9	$(1,13) \rightarrow 1$	19	$(5,5) \rightarrow (5,5)$
10	$(3,5) \rightarrow 0$	20	$(2,7) \rightarrow (3,7)$

5 КУБІЧНЕ ЧИСЛЕННЯ. МОДЕЛІ ЕЛЕМЕНТІВ ТА ОСНОВНІ ОПЕРАЦІЇ

Мета: побудова кубічних покриттів комбінаційних схем і застосування операцій кубічного числення.

Кубічне числення є математичним апаратом, використовуваним при синтезі та аналізі в більшості сучасних систем автоматизованого проєктування (САПР) цифрових схем.

Кубічне подання логічних функцій найбільш придатне для машинних методів їх аналізу, оскільки дає змогу компактно зображувати логічні функції великої кількості змінних. Ця властивість визначила найбільше поширення кубічного подання логічних функцій під час створення САПР цифрової апаратури. Саме тому кубічне подання логічних функцій заслуговує на докладніший розгляд.

Для подання логічних функцій у кубічній формі використовують чотирисимвольний теоретико-множинний алфавіт, заснований на двох логічних примітивах 0 та 1, тобто кубічний алфавіт можна уявити як $A = \{0, 1, U, X\}$. Зазвичай у теоретико-множинному поданні символ U асоційований із поняттям «порожньо». Символ X цьому алфавіті можна розглядати як універсум, тобто $X = 0 = 1 = U$. Надалі «порожньо» та U («unknown state») розглядають як синоніми.

Приклади виконання завдань

Завдання 5.1

Скласти кубічне покриття для елементів 3І, ЗАБО, 3І-НІ, ЗАБО-НІ, 2XOR.

На рисунку 5.1 наведено приклад кубічних покриттів для елементів 3І, ЗАБО, 3І-НІ, ЗАБО-НІ, 2XOR.

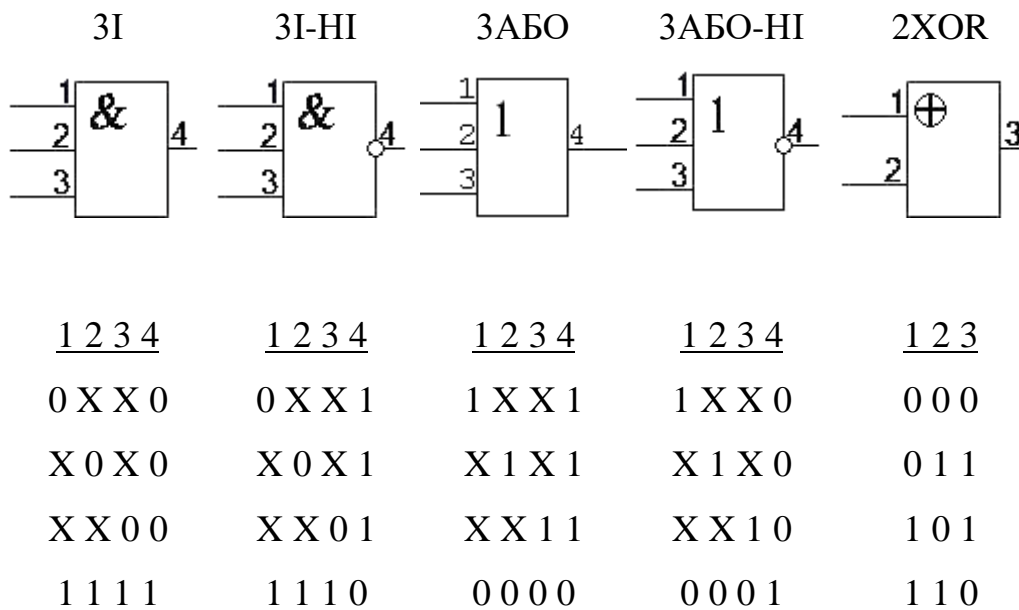


Рисунок 5.1

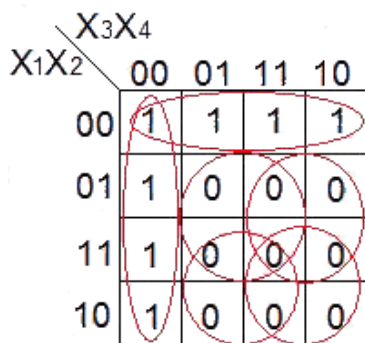
Слід зазначити, що структура кубічних покриттів логічних елементів регулярна, тобто зі збільшенням кількості входів логічних елементів структура кубічних покриттів не змінюється.

Завдання 5.2

На основі заданого кубічного покриття C^1 побудувати C^0 , скласти схему без інверсій на входах і промоделювати заданий вхідний набір.

Дано: $C^1 = \begin{Bmatrix} 00XX \\ XX00 \end{Bmatrix}$, вхідний набір $T = \{ 0 0 1 1 \}$.

Виходячи з припущення, що розглянута логічна функція є повністю визначеною, одним із найпростіших способів отримання C^0 є заповнення карти Карно «1» на підставі C^1 , після чого порожні клітинки заповнюють «0». Потім виконують мінімізацію за «0» (обведення груп нулів) і одержання на цій основі кубічного покриття C^0 (рисунок 5.2).



Чотири обведення по чотири нулі дають чотири куби другого рангу, які складають покриття C^0 :

$$C^0 = \left\{ \begin{array}{l} 1XX1 \\ 1X1X \\ X1X1 \\ X11X \end{array} \right\} \quad (5.1)$$

Рисунок 5.2

Для побудови схеми без інверсій на входах на підставі C^1 будують аналітичний вираз функції у формі ДНФ (кожному кубу C^1 відповідає терм ДНФ, координати куба, що не дорівнюють X, утворюють букви терма, і над ними ставлять інверсії, якщо на відповідних координатах стоїть «0»). До виразу C^1 застосовують правило де Моргана (рисунок 5.3)

$$Y = \overline{X_1} \overline{X_2} \vee \overline{X_3} \overline{X_4} \implies Y = \overline{\overline{\overline{\overline{\overline{X_1} \overline{X_2} \vee \overline{X_3} \overline{X_4}}}}} = \overline{\overline{\overline{X_1} \overline{X_2}} \& \overline{\overline{X_3} \overline{X_4}}} = \overline{X_1 \vee X_2} \& \overline{X_3 \vee X_4} \quad (5.2)$$

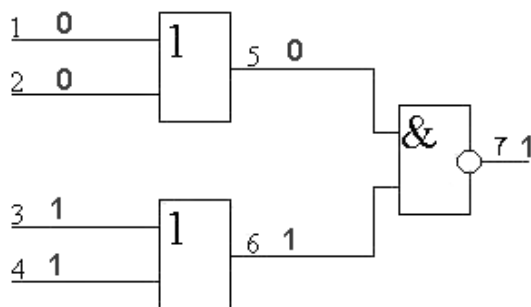


Рисунок 5.3

Синхронне моделювання заданого вхідного набору являє собою одержання значень сигналів на всіх нумерованих лініях схеми. У результаті утворений вектор

1	2	3	4	5	6	7
0	0	1	1	0	1	1

Завдання 5.3

Функціональний елемент **Комутатор 4 х 1 (мультиплексор) із прямим виходом.**

Комутатор (друга назва – мультиплексор) 4 х 1 являє собою функціональний елемент, що має чотири інформаційних входи D0, D1, D2, D3, два керуючі A, B та один вихід. Кількість інформаційних і керуючих входів задано співвідношенням $n = 2^m$, де n — кількість інформаційних входів, m — кількість керуючих (адресних) входів. Інформація передається на вихід Y з того інформаційного входу, чий номер у двійковому еквіваленті заданий значеннями 0 і 1 на керуючих входах A і B. Якщо A = B = 0, то на вихід Y подаватиметься значення входу D0, якщо A = 0 і B = 1 — значення D1, якщо A=1 і B=0 — значення D2, якщо A = B = 1 — значення D3 (рисунок 5.4).

	D0	D1	D2	D3	A	B	Y
	0	x	x	x	0	0	0
	1	x	x	x	0	1	1
	x	0	x	x	1	0	0
	x	1	x	x	1	1	1
	x	x	0	x	1	0	0
	x	x	1	x	1	1	1
	x	x	x	0	1	1	0
	x	x	x	1	1	1	1
	1	1	1	1	x	x	0
	0	0	0	0	x	x	1
	0	0	x	x	0	x	0
	1	1	x	x	0	x	1
	x	x	0	0	x	1	0
	x	x	1	1	x	1	1

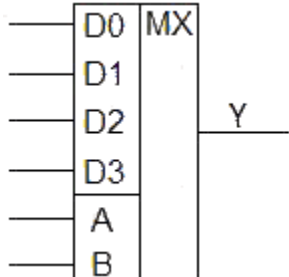


Рисунок 5.4

Варіанти завдань для самостійного виконання наведено в таблиці 5.1.

Таблиця 5.1

Варіант	Завдання 5.1	Завдання 5.2		Завдання 5.3
	Елемент	Кубічне покриття C^1	Вхідний набір T	Тип функціонального елемента
1	2	3	4	5
1	4І	{ 0XX0, X00X }	$T = \{ 0\ 1\ 1\ 0 \}$	Комутатор 4 в 1 з інверсними входами, прямим виходом
2	4АБО	{ 1XX1, X11X }	$T = \{ 1\ 0\ 0\ 1 \}$	Комутатор 4 в 1 з інверсними входами, інверсним виходом
3	4І-НІ	{ X0X0, 0X0X }	$T = \{ 0\ 1\ 0\ 1 \}$	Дешифратор 2 в 4 із прямими виходами
4	4АБО-НІ	{ X1X1, 1X1X }	$T = \{ 1\ 0\ 1\ 0 \}$	Дешифратор 2 в 4 з інверсними виходами
5	3XOR	{ 0XX0, X00X }	$T = \{ 0\ 0\ 0\ 0 \}$	Шифратор 4 в 2 із прямими виходами
6	5І	{ 1XX1, X11X }	$T = \{ 1\ 1\ 1\ 1 \}$	Шифратор 4 в 2 з інверсними виходами
7	5АБО	{ 00XX, 00XX }	$T = \{ 0\ 0\ 1\ 1 \}$	Демультіплексор 1 в 4 з інверсними входами, прямим виходом
8	5І-НІ	{ 11XX, XX11 }	$T = \{ 1\ 1\ 0\ 0 \}$	Демультіплексор 1 в 4 з інверсними входами, інверсним виходом
9	5АБО-НІ	{ 0XX0, X00X }	$T = \{ 0\ 1\ 1\ 0 \}$	Комутатор 8 в 1 з інверсними входами, прямим виходом

Продовження таблиці 5.1

1	2	3	4	5
10	4XOR	{ 1XX1, X11X }	$T = \{ 1\ 0\ 0\ 1 \}$	Комутатор 8 в 1 з інверсними входами, інверсним виходом
11	6I	{ X0X0, 0X0X }	$T = \{ 0\ 1\ 0\ 1 \}$	Дешифратор 3 в 8 із прямими виходами
12	6АБО	{ X1X1, 1X1X }	$T = \{ 1\ 0\ 1\ 0 \}$	Дешифратор 3 в 8 з інверсними виходами
13	6I-НІ	{ 0XX0, X00X }	$T = \{ 0\ 0\ 0\ 0 \}$	Шифратор 8 в 3 із прямими виходами
14	6АБО-НІ	{ 1XX1, X11X }	$T = \{ 1\ 1\ 1\ 1 \}$	Шифратор 8 в 3 з інверсними виходами
15	2I-НІ	{ 00XX, 00XX }	$T = \{ 0\ 0\ 1\ 1 \}$	Демультимплексор 1 в 8 з інверсними входами, прямим виходом
16	5XOR	{ 11XX, XX11 }	$T = \{ 1\ 1\ 0\ 0 \}$	Демультимплексор 1 в 8 з інверсними входами, інверсним виходом
17	2АБО-НІ	{ 0XX0, X00X }	$T = \{ 0\ 1\ 1\ 0 \}$	Комутатор 16 в 1 з інверсними входами, прямим виходом

6 АСИНХРОННЕ ДВІЙКОВЕ МОДЕЛЮВАННЯ

Мета: асинхронне двійкове моделювання комбінаційної схеми з заданими затримками і трійкове моделювання; побудова часової діаграми (wafeform); визначення наявності ризиків збою.

Моделювання цифрової комбінаційної схеми

Однією з систем компілятивного асинхронного двійкового моделювання є підсистема моделювання в САПР Active_HDL, у документації до цієї системи є досить докладний опис так званого «умовного асинхронного моделювання за станом подій».

Асинхронне моделювання полягає в обчисленні сигналів на виходах усіх логічних елементів схеми відповідно до її моделі. При асинхронному моделюванні потрібно визначити не тільки стан на виході компонента (0 або 1), але й час, коли відбулася зміна. Для цього вибирають часовий крок τ як максимальний загальний дільник часів затримок компонентів схеми та керуючих впливів. Час моделювання T розбивають на $n = T/\tau$ інтервалів і відповідно до розрахункових значень рівнів у всіх вузлах схеми в кінці кожного інтервалу.

Слід зазначити, що співвідношення являє собою набір явних формул, що дають змогу розрахувати стан схеми в будь-який момент часу, проте для складних схем, які зазвичай і зустрічаються на практиці, такий розрахунок дуже трудомісткий. Для скорочення витрат за асинхронного моделювання схем використовують описаний вище алгоритм.

У системах асинхронного двійкового моделювання час моделювання змінюється не поступово з задалегідь заданим кроком τ , а в моменти виникнення подій. Ці моменти визначені затримками логічних елементів, які різні.

Щодо інтерпретативних систем моделювання, то асинхронне моделювання має ряд особливостей.

В інтерпретативних системах моделювання мікротактів визначається мінімальною затримкою одного з примітивних елементів схеми. Усі інші затримки визначають як кратні мінімальній. Час моделювання одного набору схеми називається тактом (Т). Такт дорівнює кількості мікротактів на шляху максимальної довжини в схемі.

Приклад виконання завдання (таблиця 6.1)

Таблиця 6.1

Варіант	Змінні	Логічна функція	Форма подання	Пара наборів
Приклад	3	$Y(2, 5, 6, 7) = 1$	ДНФ	110 та 111

Аналізують схему на наявність у ній ризиків збою в такій послідовності.

1 Мінімізація булевої функції та отримання рівняння у формі ДНФ (рисунок 6.1)

Карта Карно

		x2x3			
		00	01	11	10
x1	0	0	0	0	1
	1	0	1	1	1

Кубічне покриття

$$C^0 = \{X00\}$$

$$C^0 = \{0X1\}$$

$$C^0 = \{X10\}$$

$$C^0 = \{1X1\}$$

Булево рівняння у формі ДНФ

$$y = x_1x_3 \vee x_2\bar{x}_3, \Delta=1.$$

Рисунок 6.1

2 Побудова цифрової схеми у вибраному базисі

На основі булевого рівняння у формі ДНФ будують цифрову схему для заданої логічної функції в базисі **I-АБО-НІ**. Затримки всіх елементів (дельта) приймемо рівними 1 ($\Delta = 1$). На рисунку 6.2 на кожному вході та виході для трійкового моделювання позначимо набори А і В (можна взяти будь-які два набори) і перехідний набір А/В.

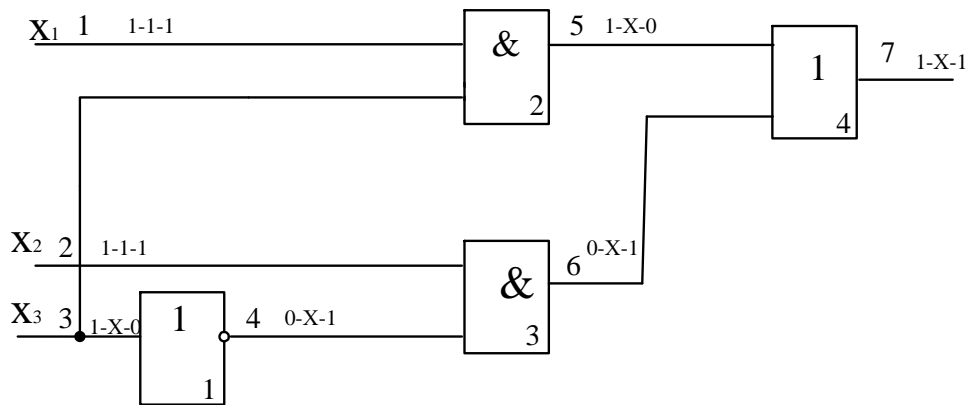


Рисунок 6.2

3 Трійкове моделювання побудованої цифрової схеми

За трійкового моделювання цифрової схеми, крім першого та другого наборів, моделюють перехідний набір, у якому на змінюваних входних координатах ставлять символ X (або U). Трійкове моделювання виконують за синхронним алгоритмом і показано в таблиці 6.2.

Таблиця 6.2

	1	2	3	4	5	6	7
A	1	1	1	0	1	0	1
A/В	1	1	X	X	X	X	X
B	1	1	0	1	0	1	1

На лінії 7 є комбінація 1 X 1, яка визначає можливість наявності ризику збою на лінії 7.

4 Моделювання ризиків збою в комбінаційних схемах

Завдання передбачає аналіз ризиків збою для пари наборів **111-110**. Виконаємо їхнє моделювання у прямому (111-110) і зворотному (011-111) напрямку та побудуємо часову діаграму (**waveform**) результатів моделювання вказаної пари наборів(рисунок 6.3). Слід вважати, що синхронне моделювання на обох наборах дає **1**, а трійкове вказує на наявність ризику збою.



Рисунок 6.3

Висновок. Перехід 111-110 дає ризик збою. Перехід 110-111 не змінює значення 1 на виході. У такий спосіб наявність ризику збою залежить не тільки від співвідношення затримок елементів, а і порядку проходження наборів один за одним.

Варіанти завдань для самостійного виконання наведено в таблиці 6.3.

Таблиця 6.3

Варіант	Змінні	Логічна функція	Форма подання
1	4	$f(2, 5, 6, 7, 8, 12, 13, 15) = 1$	ДНФ
2	4	$f(0, 2, 4, 7, 9, 11, 14, 15) = 1$	ДНФ
3	4	$f(0, 1, 2, 3, 6, 10, 11, 12, 14, 15) = 1$	ДНФ
4	4	$f(3, 4, 5, 6, 10, 11, 12, 13) = 1$	ДНФ
5	4	$f(1, 3, 9, 10, 11, 12, 14) = 1$	ДНФ
6	4	$f(0, 2, 3, 7, 9, 11, 13, 15) = 1$	ДНФ
7	4	$f(0, 4, 6, 8, 10, 11, 12, 14, 15) = 1$	ДНФ
8	4	$f(0, 1, 2, 3, 4, 10, 11, 12, 14, 15) = 1$	ДНФ
9	4	$f(2, 3, 5, 7, 9, 10, 11, 13, 15) = 1$	ДНФ
10	4	$f(1, 3, 4, 9, 10, 11, 12, 14) = 1$	ДНФ
11	4	$f(0, 1, 2, 5, 8, 12, 13, 14, 15) = 1$	ДНФ
12	4	$f(0, 1, 4, 5, 6, 9, 11, 12, 14) = 1$	ДНФ
13	4	$f(2, 6, 7, 8, 9, 12, 13, 15) = 1$	ДНФ
14	4	$f(1, 4, 5, 6, 9, 13, 14, 15) = 1$	ДНФ
15	4	$f(2, 4, 5, 8, 10, 11, 12, 13) = 1$	ДНФ
16	4	$f(1, 3, 5, 7, 9, 11, 12, 15) = 1$	ДНФ
17	4	$f(0, 1, 2, 3, 10, 11, 12, 14, 15) = 1$	ДНФ

7 ДЕДУКТИВНЕ МОДЕЛЮВАННЯ НЕСПРАВНОСТЕЙ

Мета: дедуктивне моделювання комбінаційної схеми на вичерпному тесті; побудова багатозначної таблиці несправностей і її мінімізування; визначення повноти отриманого тесту.

Дедуктивне моделювання несправностей

Найпростішим неявним методом моделювання несправностей є *дедуктивний* метод, орієнтований на обробку схем вентильного рівня логічного (І, АБО, НІ) або універсального базисів. Для зазначених типів примітивних елементів (ПЕ) розроблено формули отримання списків несправностей на виходах ПЕ залежно від виду двійкових наборів на входах ПЕ. Алгоритм дедуктивного моделювання несправностей складається з таких кроків.

- 1 Побудова структурно-функціональної моделі схеми.
- 2 Справне моделювання заданого вхідного набору.
- 3 Складання початкових списків несправностей на зовнішніх входах модельованої схеми. До початкових списків входять константні несправності, визначені інверсними від справних значень сигналів на зовнішніх входах.

4 Вибирають один за одним ПЕ схеми, і для кожного з них виконують формування вихідних списків несправностей на підставі результатів справного моделювання і дедуктивних формул для елементів логічного базису (І, АБО, І-НІ, АБО-НІ). Слід зазначити, що інверсія не впливає на вигляд формул дедуктивного моделювання. На рисунку 7.1 наведено дедуктивні формули одержання вихідних списків несправностей для елементів І та АБО.

Зазначимо, що до вихідного списку несправностей для ПЕ завжди додана власна несправність виходу, що являє собою інверсне значення від справного сигналу на виході цього ПЕ.

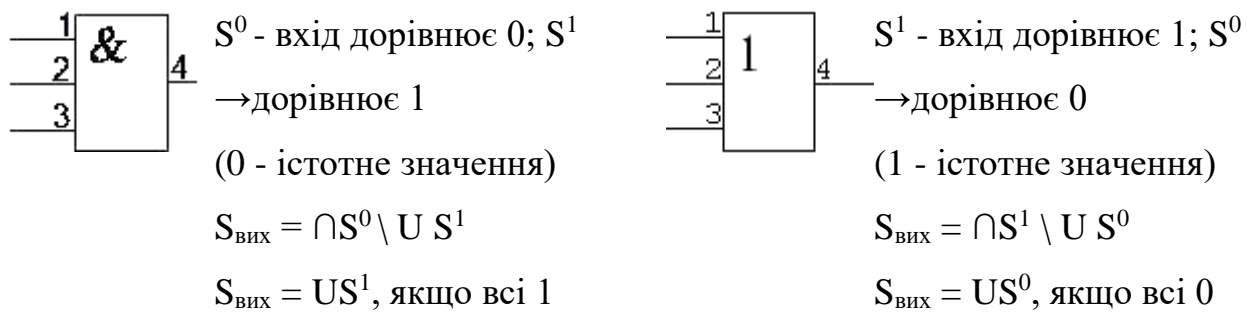


Рисунок 7.1

5 Пункт 4 виконують доти, поки не будуть складені списки, що перевіряють несправності для всіх ПЕ.

6 На підставі результуючого списку несправностей на зовнішньому виході (виходах) схеми формують рядок таблиці несправностей.

Дедуктивний алгоритм моделювання несправностей ефективно працює для схем вентильного рівня, а також зручний при вивченні ручних методів моделювання несправностей. Для схем вищого рівня ієрархії складання й використання дедуктивних формул досить трудомістке.

Приклад виконання завдання (таблиця 7.1)

Таблиця 7.1

Варіант	Змінні	Логічна функція	Форма подання
Приклад	3	$Y(2, 5, 6, 7) = 1$	ДНФ

1 Підготовка схеми до моделювання несправностей

Побудуємо цифрову схему для заданої логічної функції $Y(2, 5, 6, 7) = 1$ у формі ДНФ із використанням карт Карно (рисунок 7.2).

Виконаємо мінімізацію.

Карта Карно

Кубічне покриття

Булеве рівняння у формі
ДНФ

	x2x3			
	00	01	11	10
x1				
0	0	0	0	1
1	0	1	1	1

$$C^0 = \{X00\}$$

$$C^0 = \{0X1\}$$

$$y = x_1x_3 \vee x_2\bar{x}_3$$

Рисунок 7.2

Для отриманої схеми (рисунок 7.3) виконаємо ранжирування шляхом нумерації її ліній, а потім виконаємо дедуктивне моделювання несправностей для вичерпного тесту - $2^n = 8$ наборів.

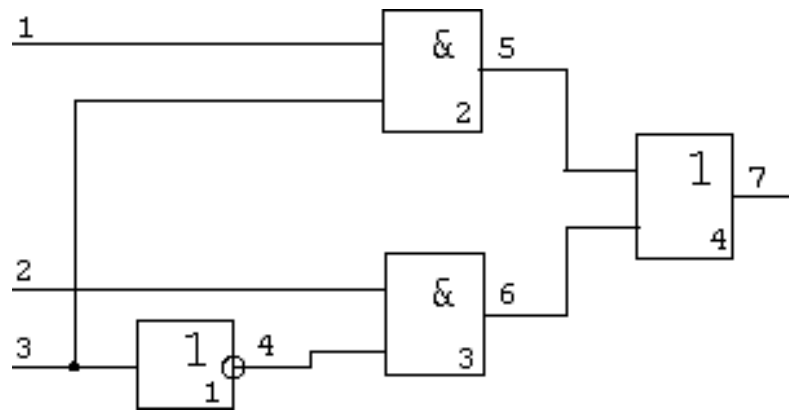


Рисунок 7.3

2 Результати моделювання несправностей для вичерпного тесту

(рисунок 7.4)

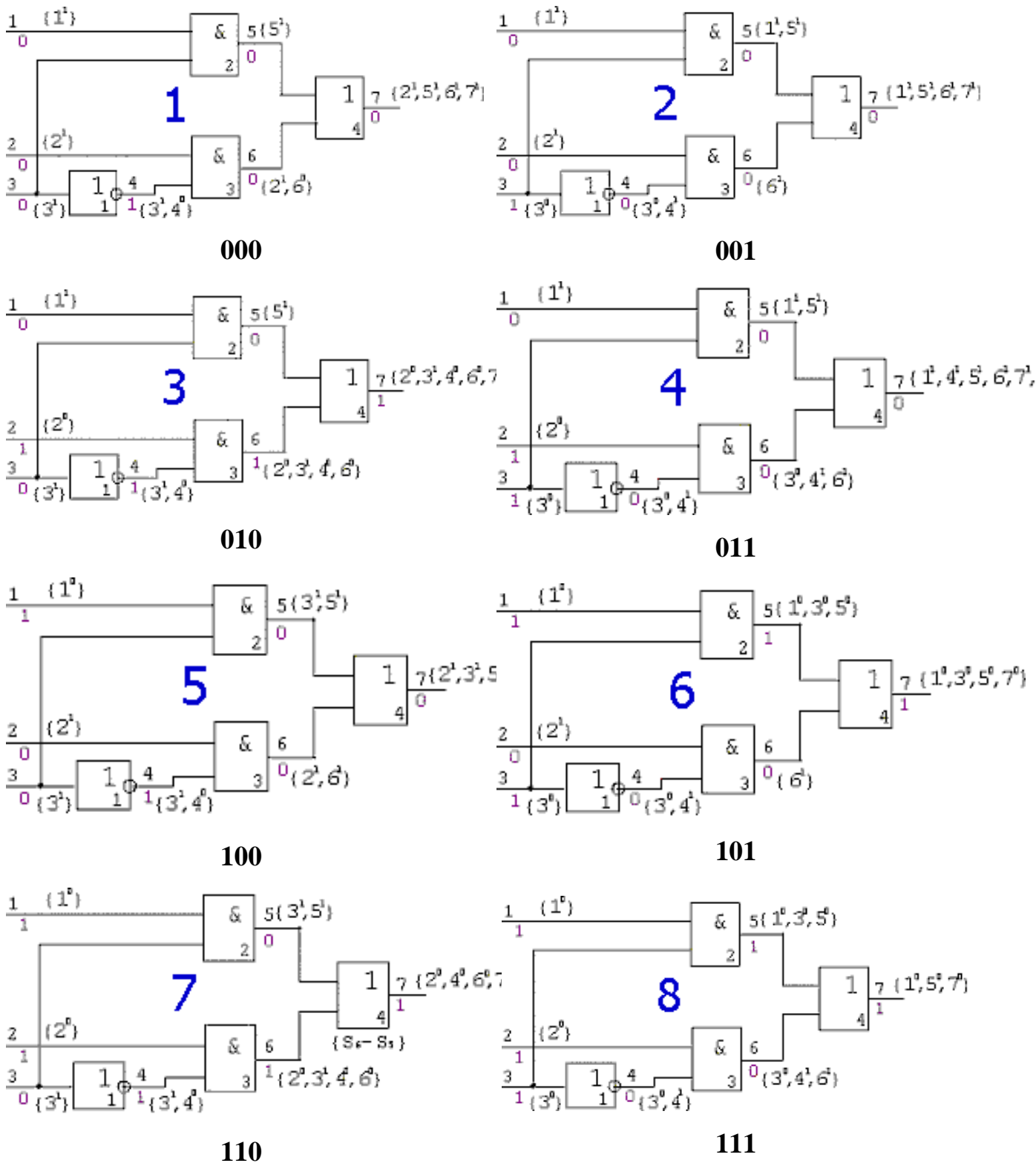


Рисунок 7.4

3 Таблиця несправностей для вичерпного тесту

Вичерпний тест – це 2^n наборів, де n – кількість входів схеми, що перевіряють, спосіб генерації підсумовування ($A = A + 1$) на n -розрядному лічильнику.

На підставі результатів моделювання несправностей на всіх наборах заповнюємо багатозначну (поєднану або суміщену) таблицю несправностей (ТН) (таблиця 7.2). Кількість рядків цієї ТН дорівнює кількості тестових наборів (8), а кількість стовпців - кількості ліній схеми (7). У комірці таблиці стоїть 1, якщо на тестовому наборі (у відповідному рядку) на лінії, що розглядають (відповідний стовпець) перевіряється несправність «константа 1»; у комірці таблиці стоїть 0, якщо на тестовому наборі (рядок) на лінії (стовпець) перевіряється несправність «константа 0». Якщо жодна несправність на лінії не перевіряється цим тестовим набором, то в комірці таблиці стоїть «-».

На підставі результату дедуктивного моделювання одного набору заповнюють один рядок суміщеної ТН, що було показано під час розгляду прикладу дедуктивного моделювання.

Таблиця 7.2

		1	2	3	4	5	6	7
1	000	-	1	-	-	1	1	1
2	001	1	-	-	-	1	1	1
3	010	-	0	1	0	-	0	0
4	011	1	-	0	1	1	1	1
5	100	-	1	1	-	1	1	1
6	101	0	-	0	-	0	-	0
7	110	-	0	-	0	-	0	0
8	111	0	-	-	-	0	-	0

ТН має відповідати таким умовам:

- 1) виявлення – у кожному стовпці ТН має стояти хоча б одна 1;
- 2) розрізнення – у ТН не має бути двох однакових стовпців;
- 3) мінімальність – при видаленні з ТН будь-якого рядка порушуватиметься умова виявлення.

Для реалізації умов виявлення кожної з несправностей у кожному стовпці мають бути 0 й 1. У такий спосіб тест 100 % повноти повинен обов'язково мати у всіх стовпцях хоча б по одному 0 й 1 (принцип виявлення). Для одержання мінімального тесту видалимо несуттєві рядки 1, 2, 7, 8 (у таблиці 7.3 позначені рожевим кольором).

Таблиця 7.3 – Таблиця несправностей для мінімізованого тесту

		1	2	3	4	5	6	7
3	010	-	0	1	0	-	0	0
4	011	1	-	0	1	1	1	1
5	100	-	1	1	-	1	1	1
6	101	0	-	0	-	0	-	0

Висновок. У кожному стовпці багатозначної (поєднаної) ТН є 0 й 1, отже, повнота тесту – 100 %. Жоден із рядків ТН не можна видалити без порушення принципу виявлення, отже, тест мінімальний.

Варіанти завдань для самостійного виконання наведено в таблиці 7.4.

Таблиця 7.4

Варіант	Змінні	Логічна функція	Форма подання
1	4	$f(2, 5, 6, 7, 8, 12, 13, 15) = 1$	ДНФ
2	4	$f(0, 2, 4, 7, 9, 11, 14, 15) = 1$	ДНФ
3	4	$f(0, 1, 2, 3, 6, 10, 11, 12, 14, 15) = 1$	ДНФ
4	4	$f(3, 4, 5, 6, 10, 11, 12, 13) = 1$	ДНФ
5	4	$f(1, 3, 9, 10, 11, 12, 14) = 1$	ДНФ
6	4	$f(0, 2, 3, 7, 9, 11, 13, 15) = 1$	ДНФ
7	4	$f(0, 4, 6, 8, 10, 11, 12, 14, 15) = 1$	ДНФ
8	4	$f(0, 1, 2, 3, 4, 10, 11, 12, 14, 15) = 1$	ДНФ
9	4	$f(2, 3, 5, 7, 9, 10, 11, 13, 15) = 1$	ДНФ
10	4	$f(1, 3, 4, 9, 10, 11, 12, 14) = 1$	ДНФ
11	4	$f(0, 1, 2, 5, 8, 12, 13, 14, 15) = 1$	ДНФ
12	4	$f(0, 1, 4, 5, 6, 9, 11, 12, 14) = 1$	ДНФ
13	4	$f(2, 6, 7, 8, 9, 12, 13, 15) = 1$	ДНФ
14	4	$f(1, 4, 5, 6, 9, 13, 14, 15) = 1$	ДНФ
15	4	$f(2, 4, 5, 8, 10, 11, 12, 13) = 1$	ДНФ
16	4	$f(1, 3, 5, 7, 9, 11, 12, 15) = 1$	ДНФ
17	4	$f(0, 1, 2, 3, 10, 11, 12, 14, 15) = 1$	ДНФ

Список літератури

- 1 Проектування та діагностика комп'ютерних систем та мереж: навч. посіб. / М. Ф. Бондаренко, Г. Ф. Кривуля, В. І. Хаханов та ін. Київ: НМЦ ВО, 2000. 306 с.
- 2 Хаханов В. І. Технічна діагностика елементів та вузлів персональних комп'ютерів: навч. посіб. Київ: ИСМО, 1997. 308 с.
- 3 Сафарбаков А. М., Лук'янов А. В., Пахомов С. В. Основи технічної діагностики: навч. посіб. Житомир: ЖГУПС, 2006. 216 с.
- 4 Стецюк А. Є., Бобровніков Я. Ю. Основи технічної діагностики. Теорія розпізнавання: навч. посіб. Кривий Ріг: Вид-во ДВГУПС, 2012. 69 с.
- 5 Глазунов Л. П., Смірнов О. М. Проектування технічних систем діагностування. Львів: Енергоатом, 2003. 168 с.
- 6 Мозгалевський А. В., Койда О. М. Запитання проектування систем діагностування. Львів: Енергоатом, 2007. 112 с.
- 7 Іоффе М. М. Оцінка можливості сигнатурного аналізу виявляти помилки заданої кратності в двійковій послідовності. *Автоматика та телемеханіка*. 2015. № 12. С. 110-115.
- 8 Ярмолик В. М. Про достовірність контролю двійкових послідовностей даних методом сигнатурного аналізу. *Електронне моделювання*. 2009. Т. 7. № 6. С. 51-54.
- 9 Стандарт граничного сканування IEEE 1149.1. URL: <http://standards.ieee.org/findstds/standard/1149.1-2001.html/> (дата звернення 5.05.2022).
- 10 Стенбрідж Дж. Успіхи та перспективи граничного сканування. *Електронні компоненти*. 2013. № 1. С. 22-26.
- 11 Городецький А. В., Курилан Л. В. Тестопридатне проектування схем для граничного сканування. *Виробництво електроніки*. 2008. № 24. С. 25-26.
- 12 Платунов А. Є., Постніков Н. П., Чистяків А. Г. Механізми граничного сканування в неоднорідних мікропроцесорних системах. *ChipNews*. 2000. № 3. С. 45-48.

МЕТОДИЧНІ ВКАЗІВКИ

до практичних занять, лабораторних робіт і курсового проектування з дисциплін
*«ТЕХНІЧНА ДІАГНОСТИКА ТА ТЕСТОПРИДАТНІСТЬ ЦИФРОВИХ
ПРИСТРОЇВ, КОМП'ЮТЕРНИХ СИСТЕМ, МЕРЕЖ ТА ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ»,
«ДІАГНОСТУВАННЯ ТА ТЕСТУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ»*

Відповідальний за випуск Клименко Л. А.

Редактор Ібрагімова Н. В.

Підписано до друку 08.05.2024 р.

Умовн. друк. арк. 3,5. Тираж . Замовлення № .

Видавець та виготовлювач Український державний університет залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха,7.

Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.