

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Моделі тестопридатності реконфігурованих інтерактивних комп'ютерних
мереж на програмованих функціональних модулях
(тема)

Виконав: студент 2 курсу, групи СКСм-22-2

Лобойченко Д.А.

(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма

Спеціалізовані комп'ютерні системи

(повна назва освітньої програми)

Керівник роботи

доц. Шкіль О.С.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Чумаченко С.В.

(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки


Рівень вищої освіти другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія
(шифр і назва)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 
(підпис)

« 02 » 09 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студенту Лобойченко Данилі Анатолійовичу
(прізвище, ім'я, по батькові)

Тема роботи (проекту) Моделі тестопридатності реконфігурованих інтерактивних комп'ютерних мереж на програмованих функціональних модулях

Testability Models of Reconfigurable Interactive Computing Networks on Programmable Functional Modules

2. Термін подання студентом роботи 10.01.2024

3. Вихідні дані до роботи (проекту) _____

Клітинні автомати

Інтерактивні комп'ютерні мережі

Реконфігуровані комп'ютерні мережі

ПЛІС FPGA

Перелік питань, що потрібно опрацювати у роботі _____

Методи проектування реконфігуруючих інтерактивних комп'ютерних мереж.

Класифікація методів проектування реконфігуруючих інтерактивних комп'ютерних мереж.

Верифікація HDL-моделей на програмуваних функціональних модулях.

Методи діагностування реконфігурованих систем.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 22 слайди

6. Консультанти розділів роботи (проекту)


| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|----------------------|---|---|------|
| | | підпис | дата |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання 02.09.2023

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи (проекту) | Термін виконання етапів роботи | Примітка |
|---|---|--------------------------------|----------|
| 1 | Видача теми проекту, узгодження і затвердження теми | 02.09.2023-08.09.2023 | |
| 2 | Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів | 09.09. 2023-20.09. 2023 | |
| 3 | Методи проектування реконфігурованих інтерактивних комп'ютерних мереж | 21.09.2023-10.10.2023 | |
| 4 | Класифікація методів проектування реконфігурованих ІКМ | 10.10. 2023-31.10.2023 | |
| 5 | Методи ре конфігурації одновимірних та двовимірних ІКМ | 01.11. 2023-20.11. 2023 | |
| 6 | Методи діагностування реконфігурованих ІКМ | 20.11. 2023-10.12. 2023 | |
| 7 | Проведення ДЕ | 11.12. 2023-25.12. 2023 | |
| 8 | Оформлення пояснювальної записки | 25.12. 2023-10.01. 2024 | |
| 9 | Захист проекту | 12.01. 2024-25.01. 2024 | |
| | | | |

Студент 
(підпис)

Керівник роботи (проекту) 
(підпис)

доц.Шкіль О.С
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 74 сторінки, 27 рисунків, 16 джерел за переліком посилань.

ІНТЕРАКТИВНА КОМП'ЮТЕРНА МЕРЕЖА, ФУНКЦІОНАЛЬНИЙ МОДУЛЬ, СИСТОЛІЧНІ СТРУКТУРИ, КОМІРКА, ТЕСТОВИЙ ГРАФ, ДІАГНОСТИЧНИЙ ЕКСПЕРИМЕНТ, МОВА ОПИСУ АПАРАТУРИ

Метою кваліфікаційної роботи є проектування реконфігурованих інтерактивних комп'ютерних мереж (ІКМ) на програмованих функціональних модулях (ФМ), а також методик проведення діагностичних експериментів над моделями зазначених мереж.

Проведений аналіз існуючих методик проведення діагностичних експериментів над кінцевими автоматами, проаналізовані методи підвищення тестопридатності кінцевих автоматів, розроблені методики проведення діагностичних експериментів над кінцевими автоматами в умовах введення апаратної надлишковості в моделі автоматів. Розроблені методики реалізовані в САПР на основі мов опису апаратури. Запропоновано процедуру модифікації структури функціональних комірок мережі шляхом введення додаткового входу, що спрощує процедуру синтезу перевіряючих тестів для ІКМ із бічними виходами, що спостерігаються, з метою покращення показників керованості ІКМ у процесі діагностичного експерименту.

Розроблено процедуру побудови діагностичного експерименту для ІКМ із виходами, що спостерігаються, у якій автоматна модель ФМ мережі є сильнозв'язним автоматом і має відрізняючу послідовність. Показано, що така мережа є тестопридатною.

ABSTRACT

The explanatory note contains: 74 pages, 27 figure, 16 sources according to the list of links.

INTERACTIVE COMPUTER NETWORK, FUNCTIONAL MODULE, SYSTOLIC STRUCTURES, CELL, TEST GRAPH, DIAGNOSTIC EXPERIMENT, HARDWARE DESCRIPTION LANGUAGE

The purpose of the qualifying work is the design of reconfigured interactive computer networks (ICN) on programmable functional modules (FM), as well as methods for conducting diagnostic experiments on models of these networks.

An analysis of existing methods for conducting diagnostic experiments on finite state machines (FSM) has been carried out, methods for increasing the testability of FSM have been analyzed, and methods have been developed for conducting diagnostic experiments on FSM under the conditions of introducing hardware redundancy into FSM models. The developed methods are implemented in CAD based on hardware description languages. A procedure has been proposed for modifying the structure of the functional cells of the network by introducing an additional input, which simplifies the procedure for synthesizing verification tests for ICN with observable side outputs in order to improve the controllability of the ICN in the process of a diagnostic experiment.

A procedure has been developed for constructing a diagnostic experiment for ICN with observable outputs, in which the automaton model of the FM network is a strongly connected FSM and has a distinctive sequence. It is shown that such a network is testable.

ЗМІСТ

| | |
|--|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ..... | 8 |
| ВСТУП..... | 10 |
| 1 ОДНОРІДНІ ОБЧИСЛЮВАЛЬНІ СИСТЕМИ ТА ЇХ ДІАГНОСТУВАННЯ..... | 12 |
| 1.1 Систолічні структури | 12 |
| 1.2 Тестування систолічних матриць | 18 |
| 1.3 Проектування тестопридатних систолічних матриць процесорних елементів..... | 22 |
| 2 ТЕХНОЛОГІЇ РОЗВИТКУ РОЗПОДІЛЕНИХ КОМП'ЮТЕРНИХ СИСТЕМ НА ПРОГРАМОВАНИХ ФУНКЦІОНАЛЬНИХ МОДУЛЯХ..... | 25 |
| 2.1 Аналіз структур та тенденцій розвитку інтерактивних комп'ютерних мереж | 25 |
| 2.2 Синтез перевіряючої послідовності за тестовим графом функціональної комірки ІКМ | 33 |
| 2.3 Концепція С-тестованості інтерактивних комп'ютерних мереж..... | 37 |
| 2.4 Класифікація експериментів над кінцевими автоматами | 38 |
| 2.4.1 Послідовності автоматів та їх характеристики під час проведення експерименту | 38 |
| 2.4.2 Методи підвищення тестопридатності кінцевих автоматів | 42 |
| 3 МЕТОДИ ПРОЕКТУВАННЯ РЕКОНФІГУРОВАНИХ ІКМ НА ПРОГРАМОВАНИХ ФУНКЦІОНАЛЬНИХ МОДУЛЯХ..... | 47 |
| 3.1. Синтез одновимірних ІКМ із централізованим управлінням реконфігурацією | 47 |
| 3.2. Синтез ІКМ з розподіленим управлінням реконфігурацією | 57 |
| 3.2.1. Одновимірні односпрямовані ІКМ | 57 |
| 3.2.2. Двовимірні ІКМ з розподіленим управлінням реконфігурацією..... | 59 |

| | |
|---|----|
| 3.3. Діагностування та верифікація програмних модулів HDL-моделей ... | 66 |
| ВИСНОВКИ..... | 71 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ..... | 73 |
| ДОДАТОК А Графічна частина кваліфікаційної роботи | 75 |
| ДОДАТОК Б Листинги HDL-моделі | 92 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

| | |
|---------|---|
| БВВ – | блоки вводу/виводу; |
| БПСОД– | багатопроцесорна система обробки даних; |
| ДЕ – | діагностичний експеримент; |
| ДІ-ІВ – | діагностична інфраструктура з інтелектуальними властивостями; |
| ДП – | діагностичний пристрій; |
| ІК – | інтерактивна комірка; |
| ІКМ – | інтерактивна комп'ютерна мережа; |
| ІМС – | інтегральна мікросхема; |
| ІМ – | інтерфейсний модуль; |
| КА – | клітинний автомат; |
| КЛБ – | конфігурований логічний блок; |
| КС – | комп'ютерна система; |
| КЧ – | комбінаційної частини |
| НВІС – | надвеликі інтегральні схеми; |
| ОД – | об'єкт діагностування; |
| ОІКМ – | одновимірна ІКМ; |
| ОП – | оперативна пам'ять; |
| ПЕ – | процесорний елемент; |
| ПЛІС – | програмовані логічні інтегральні схеми; |
| ППК – | програмований перемикач комірки; |
| ПЧ – | послідовностна частина |
| РІКМ – | реконфігуровані ітеративні комп'ютерні мережі; |
| РКС – | розподілена комп'ютерна система; |
| САПР – | система автоматизованого проектування; |
| СоК – | система на одному кристалі (SoC); |

- СМКЕ – систолічні матриці комбінаційних елементів;
- СМПЕ – систолічні матриці процесорних елементів;
- СП – синхронізуюча послідовність;
- СТД – система технічного діагностування;
- ТГ – тестовий граф;
- ТД – тестове діагностування;
- ТПВ – таблиця переходів-виходів;
- ФМ – функціональний модуль;
- ХП – характеристична послідовність;
- ЦП – цифровий пристрій;
- DCST – Discrete Cosine Transform (дискретне косинусне перетворення);
- IP-core – intellectual property core (ІР-ядра, готові блоки для проектування ПЛІС).

ВСТУП

Розвиток та впровадження наноелектронних технологій у процес створення сучасних інформаційно-керуючих та КС, широке використання ПЛІС, сигнальних процесорів, мікроконтролерів, мереж та систем на одному кристалі (SoC), модулів з інтелектуальними властивостями (IP-core), поява на ринку електронних послуг смарт-пристроїв, ноутбуків, мобільних пристроїв та моніторів відкривають перспективи розвитку інноваційних підходів до створення систем, що мають принципово нові архітектурні та функціональні властивості, такі як самоорганізація, реконфігурація, штучний інтелект.

Розвиток інформаційних і наноелектронних технологій визначають перспективу створення економічних, надійних і стійких до відмови КС, продуктивність яких порівнянна з продуктивністю потужних суперкомп'ютерів типу nCube, Cray, але з вартістю в десятки і сотні разів меншою.

Широке застосування елементної бази, SoC і ПЛІС при проектуванні КС дало поштовх численним дослідженням щодо створення динамічних систем реконфігурації КС, які мають властивості конструктивної однорідності та паралельності обробки даних. Це визначає концептуальні основи проектування ДІ комп'ютерних систем та їх компонент у вигляді BIST одновимірних та двовимірних ітеративних обчислювальних мереж (ІКМ) з конструктивними властивостями архітектурно-структурної однорідності, тестопридатності, самоперевіряємості, високої продуктивності та реактивності.

Об'єкт дослідження у роботі – процедури проектування реконфігурованих інтерактивних обчислювальних мереж на програмованих функціональних модулях.

Предмет дослідження – моделі, методи та процедури підвищення тестопридатності реконфігурованих інтерактивних обчислювальних мереж

на програмованих функціональних модулях на етапі проектування.

Мета роботи – розробка методів підвищення тестопридатності моделей реконфігурованих інтерактивних обчислювальних мереж на програмованих функціональних модулях а також методик проведення діагностичних експериментів над моделями зазначених мереж.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- провести аналіз існуючих методик проведення діагностичних експериментів над реконфігурованими інтерактивними обчислювальними мережами на програмованих функціональних модулях;

- проаналізувати методи підвищення тестопридатності кінцевих автоматів;

- розробити методики проведення діагностичних експериментів над кінцевими автоматами в умовах введення апаратної надлишковості в моделі реконфігурованих інтерактивних обчислювальних мереж на програмованих функціональних модулях;

- реалізувати розроблені методики в рамках САПР на основі мов опису апаратури.

1 ОДНОРІДНІ ОБЧИСЛЮВАЛЬНІ СИСТЕМИ ТА ЇХ ДІАГНОСТУВАННЯ

1.1. Систолічні структури

Систолічні архітектури виникли з бажання поєднати переваги конвеєрної та матричної обробки. Спочатку систолічні архітектури розроблялися для вузькоспеціалізованих обчислювальних систем, проте надалі виявилось можливим знайти відповідні алгоритми для досить широкого класу завдань, що дозволяють реалізувати принципи систолічної обробки.

Якщо провести паралель у фізіології, то систоли найбільше нагадують систему судин та серце, яке постійно посилає кров у всі артерії, судини та капіляри тіла.

Систолічні системи є дуже спеціалізованими обчислювачами і виробляються переважно під конкретне завдання. Фактично, завдання побудови систолічного обчислювача зводиться до побудови апаратного конвеєра, що має досить великий час отримання результату (тобто велика кількість ступенів), але при цьому порівняно невеликий час між послідовною видачою результатів, так як значна кількість проміжних значень обробляється на різних ступенях конвеєра.

Основний принцип систолічної обробки полягає в тому, щоб виконати всі стадії обробки кожного елемента даних, витягнутого з оперативної пам'яті (ОП), перш ніж знову помістити на збереження результат цієї обробки. Цей принцип реалізується систолічною матрицею процесорних елементів (ПЕ), у якій окремі ПЕ об'єднані між собою прямими та регулярними зв'язками, що утворюють конвеєри (рис. 1.1). За цими конвеєрами прокачуються операнди, тобто. кожен елемент даних витягується з пам'яті і ритмічно просувається матрицею ПЕ перш, ніж знову потрапить у пам'ять. Таким чином, може формуватися кілька потоків даних, кожен з яких утворений вихідними

операндами (елементами структури даних, що зберігається в ОП), проміжними результатами, які одержані під час виконання елементарних операцій у кожному ПЕ та елементами результуючої структури. Потоки даних синхронізовані єдиною всім ПЕ системою тактових сигналів (CLK). Під час тактового інтервалу всі ПЕ виконують коротку постійну послідовність команд.

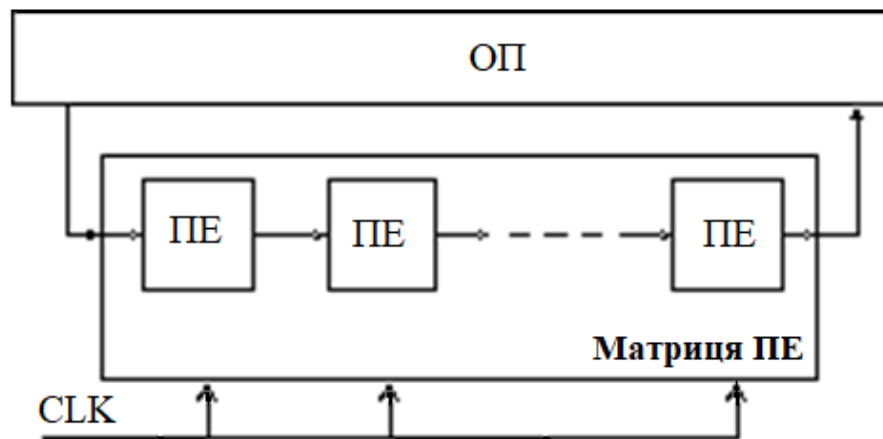


Рисунок 1.1 – Принцип систолічної обробки

Систолічна структура – це однорідне обчислювальне середовище з процесорних елементів, що поєднує в собі властивості конвеєрної та матричної обробки і має такі особливості:

- обчислювальний процес у систолічних структурах є безперервну і регулярну передачу даних від одного ПЕ до іншого без запам'ятовування проміжних результатів обчислення;
- кожен елемент вхідних даних вибирається з пам'яті одноразово і використовується стільки разів, скільки необхідно за алгоритмом; введення даних здійснюється в крайні ПЕ матриці;
- утворюють систолічну структуру ПЕ однотипні і кожен із новачків може бути менш універсальним, ніж процесори звичайних багатопроцесорних систем;

- потоки даних і керуючих сигналів мають регулярність, що дозволяє об'єднувати ПЕ локальними зв'язками мінімальної довжини;
- алгоритми функціонування дозволяють поєднати паралелізм із конвеєрною обробкою даних;
- продуктивність матриці можна поліпшити за рахунок додавання до неї певного числа ПЕ, причому коефіцієнт підвищення продуктивності при цьому лінійний.

В даний час досягнуто продуктивності систолічних процесорів близько 1000 млрд операцій/с.

Систолічна матриця – це пристрій, що складається з ідентичних комірок, локально пов'язаних один з одним, що здійснює паралельно-конвеєрну обробку даних. Систолічні матриці поділяються на систолічні матриці комбінаційних елементів (СМКЕ) та – систолічні матриці послідовностних елементів або процесорних елементів (СМПЕ). Ці пристрої відносяться до класу спеціалізованих пристроїв (призначені для виконання конкретної вузької задачі на відміну від універсальних обчислювальних засобів).

Поштовхом до розвитку таких пристроїв послужило те, що існуючі універсальні ЕОМ недостатньо забезпечені засобами обробки великих масивів даних числової або символічної природи для розпізнавання зображень, текстової інформації, мови; обробки сигналів у реальному масштабі часу. Цей чинник зумовив розвиток спеціалізованих схемних структур.

Швидкодія напівпровідникових приладів, що використовуються в ЕОМ наближається до межі, що визначається законами фізики твердого тіла, (затримка спрацьовування вентиля = $0.01\text{нс} = 10^{-11}\text{с}$). Це зумовило використання паралельних архітектур у спеціалізованих схемах.

Приклад систолічних архітектур наведено на рис. 1.2. Вони можуть бути як односпрямованими, так і дваспрямованими, рис. 1.2.а – односпрямована ортогональна одномірна, рис. 1.2.б – односпрямована

ортогональна двовимірна (існують також тривимірні), рис. 1.2.в – комірka Односпрямоваої гексагональної структури, 1.2.г – комірka односпрямованої ортогональної структури. «Систолічна» означає не просто паралельну обробку, а паралельно-конвеєрну обробку даних.

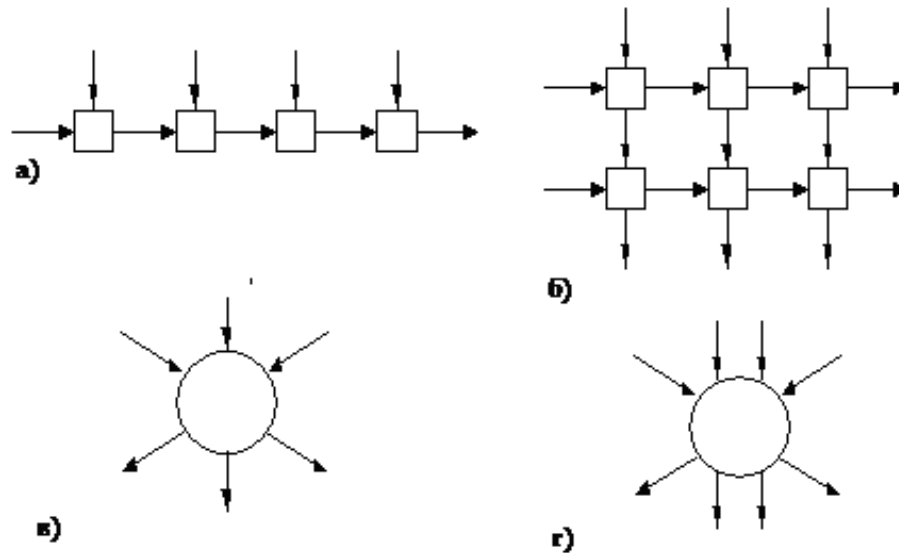


Рисунок 1.2 – Приклади систолічних архітектур

Систола (грец. *systole* – стиснення) – це стиснення передсердь і шлуночків у момент якого відбувається нагнітання крові в артерії. Разом з діастолюю (розслабленням) вони утворюють повний цикл серцевої діяльності. Таким чином, подібно до того, як кров розливається по тілу хвилями, відбувається хвилеподібна обробка інформації в матриці.

Фрагмент розв'язання задачі множення матриць систолічною структурою, $Z=AB$ (рядок на стовець) зображено на рис. 1.3. Передбачається, що спочатку вносяться в комірки значення a_{ij} , а потім зі зміщенням у часі подаються b_{ij} . Результат також знімається зі зміщенням.

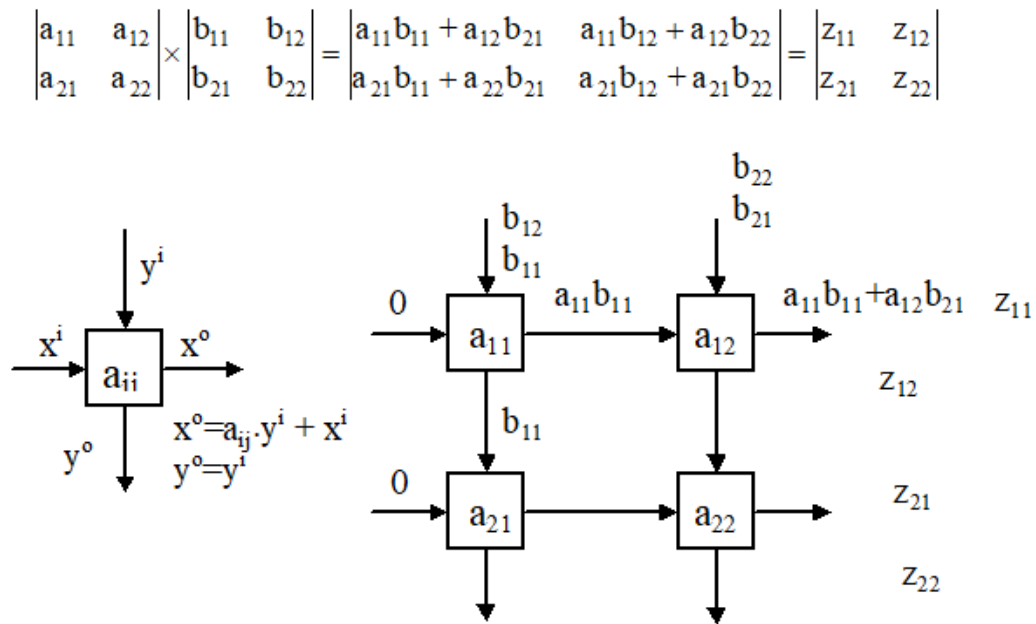


Рисунок 1.3 – Приклад множення матриць систолічною структурою

Області застосування систолічних матриць наступні.

Обробка сигналів та зображень:

– фільтрація з кінцевою та безкінечною імпульсними характеристиками;

– згортка 1 та 2 порядків, обчислення кореляції;

– інтерполяція 1 та 2 порядків та відновлення;

– дискретне перетворення Фур'є;

– кодування/декодування з корекцією помилок.

Матрична арифметика:

– множення матриць (у тому числі на вектор);

– приведення матриці до трикутного вигляду (розв'язання систем лінійних рівнянь, звернення матриці);

– обчислення власних значень;

– вирішення трикутних лінійних систем.

Нечислові програми та алгоритми:

- структури даних (стек, черги з пріоритетом, пошук, упорядкування, сортування);
- графові та геометричні алгоритми (пошук найкоротшого шляху на деревах, з'єднання компонентів, аналіз опуклих фігур);
- поліноміальні алгоритми (множення та розподіл поліномів, пошук найменших загальних дільників для поліномів);
- арифметика цілих чисел підвищеної точності (множення, розподіл, пошук найменших загальних дільників).

СМПЕ може бути побудована на дискретних елементах (процесорах). СМПЕ, СМКЕ можна реалізувати у вигляді НВІС. Реалізація у вигляді НВІС дає можливість отримання граничного відношення «продуктивність»/«вартість» (тобто максимальної продуктивності за мінімальної вартості).

Це досягається завдяки наступним перевагам систолічних матриць :

- адекватність структури матриці паралельному обчислювальному алгоритму, що реалізується нею;
- регулярність структури;
- простота та однорідність комірок;
- локальність в'язків між комірками;
- можливість модульного розширення;
- регулярність потоків команд та даних;
- паралельно-конвеєрна обробка даних;
- мінімум обмінів із зовнішнім середовищем за можливості багаторазового використання вхідних даних.

До недоліків використання систолічних матриць можна віднести:

- велика площа кристала, багато виводів;
- швидкодія матриці набагато більше швидкодії пристроїв введення-виведення.

1.2 Тестування систолічних матриць

Розглянемо тестування односпрямованої одновимірної СМКЕ.

Модель несправності – несправною може бути лише одна комірка матриці. Матриця є повністю протестованою, якщо протестовані всі комірки. Комірка є повністю протестованою, якщо протестовані всі переходи таблиці переходів/виходів.

На рис. 1.4 зображено систолічну матрицю та її комірку. Таблицю станів для неї наведено в таблиці 1.1. Для x -станів $X^I = \{x_i | 1 \leq i \leq m\}$; $X^O = \{x_k | 1 \leq k \leq m\}$; $Y^I = \{y_j | 1 \leq j \leq n\}$. Для y -станів таблиця аналогічна.

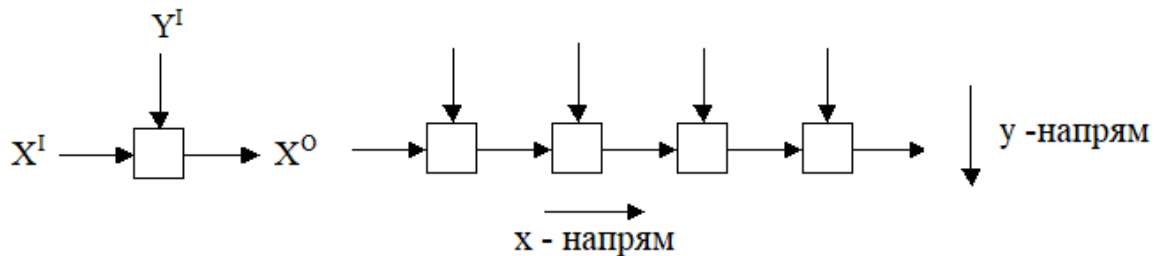


Рисунок 1.4 – Приклад систолічної матриці, що тестується

Таблиця 1.1

| Номер стану | Стан | | |
|-------------|------------|-----------|----------|
| | Позначення | Десяткове | Двійкове |
| 1 | x_1 | 0 | 00 |
| 2 | x_2 | 1 | 01 |
| 3 | x_3 | 2 | 10 |
| 4 | x_4 | 3 | 11 |

Функція комірки описується таблицею переходів S_b :

Таблиця 1.2 - S_{b1}

| | | Y^I | | | | |
|-------|-------|-------|-------|-------|-------|---|
| | | y_1 | y_2 | y_3 | y_4 | |
| X^I | x_1 | 0 | 1 | 2 | 3 | 0 |
| | x_2 | 1 | 2 | 1 | 2 | 1 |
| | x_3 | 2 | 0 | 1 | 1 | 3 |
| | x_4 | 3 | 0 | 0 | 2 | 2 |
| | | | | | | |

Кожна вхідна пара та вихідне значення становлять перехід:

$$(x_i, y_j) \rightarrow x_k, 1 \leq k \leq m.$$

Якщо є несправність в комірці, вона виявиться у вигляді несправності переходу: $(x_i, y_j) \rightarrow x_k/x^e$, де x^e – несправний стан.

Для переходу $(0,0) \rightarrow 1$ можливі такі несправності:

$$(0,0) \rightarrow 1/0, (0,0) \rightarrow 1/2, (0,0) \rightarrow 1/3.$$

Число можливих несправностей дорівнює $(m-1)$ для кожного переходу. Отже, кількість тестових векторів визначається як $T=mn(m-1)L$, де L - число комірок у матриці. Один і той самий тест не завжди може перевіряти декілька несправностей рис.1.5 (для табл. 1.2).

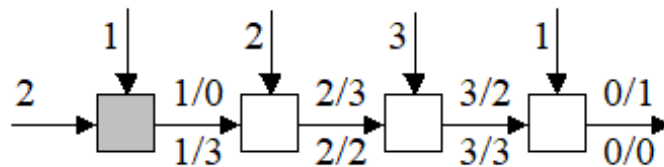


Рисунок 1.5 – Приклад несправності матриці, що тестується

У разі, якщо виникає несправність $(2,1) \rightarrow 1/3$, на виході матриці справний та несправний стан не розрізняються, тобто. $x^k = x^e$, отже, несправність маскуватиметься

Таким чином, тест T залежить від кількості комірок та кількості можливих несправностей. Крім того, для матриці з довільною функцією комірки немає гарантії покриття 100% несправностей.

Ці недоліки спричинили виникнення методу забезпечення тестопридатності матриці шляхом приведення її до виду S -тестованості.

СМКЕ є S -тестованою (константна-тестовано), якщо вона може бути протестована постійним числом тестових векторів, що не залежить від розміру матриці.

1. Тестова послідовність, яка застосовується до комірок між двома

одночасно тестованими комірками, повинна регенерувати для другої комірки такий же тестовий сигнал, що і для першої. Це умова керованості. Тестова послідовність, що застосовується до комірок між двома одночасно тестованими комірками, називається послідовністю регенерації тестового вхідного сигналу.

2. Послідовність регенерації тестового вхідного сигналу або регенеруюча послідовність повинна поширювати несправний стан до входу другої комірки і ця комірka також повинна поширювати цей несправний стан до наступної комірки, якщо перша була несправною. Це умова спостережуваності. Вона необхідна для забезпечення транспортування несправного стану до виходів, що спостерігаються.

Тестовий вхід x для першої комірки і послідовність регенерації тестового входу (y - входи) в горизонтальному напрямку називається тестовою послідовністю в горизонтальному напрямку. (Аналогічно для вертикального напрямку).

3. Тестова послідовність для тестування переходу та її довжина мають бути незалежними від несправного стану, тобто. одна і та ж послідовність повинна виявляти абсолютно всі помилки, що виникають при тестуванні переходу.

Цих трьох умов достатньо для одновимірної матриці. Для двовимірної та тривимірної матриць необхідно виконання ще однієї умови.

4. Послідовність регенерації тестового входу має бути незмінною в необхідному напрямку. Ця умова необхідна для того, щоб усі рядки і всі стовпці мали одну й ту саму горизонтальну та вертикальну послідовність регенерації тестового входу відповідно.

Функція f є добре визначеною в якомусь напрямку, скажімо, в x - напрямі, якщо для кожного $y_j \in Y$ не існує двох різних значень, $x_i, x_k \in X$, для яких виконується умова $f(x_i, y_j) = f(x_k, y_j)$.

Приклад такої функції наведено у таблиці 1.3.

Таблиця 5.3 - Sb2

| | | Y | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 |
| X | 0 | 0 | 1 | 1 | 3 |
| | 1 | 1 | 2 | 0 | 0 |
| | 2 | 2 | 3 | 2 | 1 |
| | 3 | 3 | 0 | 3 | 2 |

Матриця з повністю визначеною функцією комірки є С-тестованою (рис.1.6).

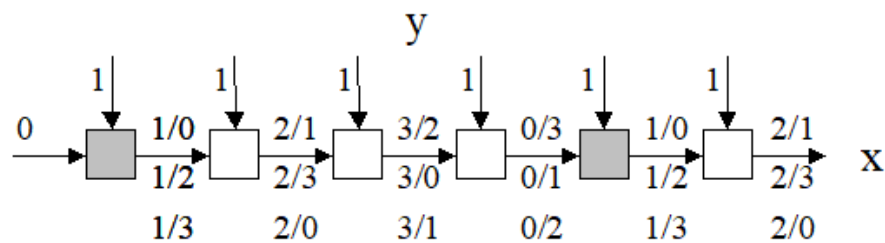


Рисунок 1.6 – Приклад С-тестованої матриці

Тестовий y-вхід $\rightarrow 1\ 1\ 1\ 1 \leftarrow$ тестовий вектор для першої (заштрихованої) комірки;

Тестовий x - вхід $\rightarrow 0\ 1\ 2\ 3 \leftarrow$ вектор моделювання;

$1\ 1\ 1\ 1 \leftarrow$ тестовий вектор для другої комірки отримано циклічним

$3\ 0\ 1\ 2$ зсувом вправо всіх розрядів тесту першої комірки та вектора моделювання;

$1\ 1\ 1\ 1 \leftarrow$ тестовий вектор для третьої комірки;

$2\ 3\ 0\ 1$

$1\ 1\ 1\ 1 \leftarrow$ тестовий вектор для четвертої комірки.

$1\ 2\ 3\ 0$

Ці тестові вектори застосовується до кожних чотирьох наступних комірок вздовж усієї матриці.

Оскільки максимально можлива довжина тестової послідовності в даному випадку дорівнює m , час тестування визначається як :

$$T=mn*m=m^2n, T \neq f(L).$$

Якщо функція не є добре визначеною, необхідно модифікувати комірку шляхом додавання додаткових станів для забезпечення її C-тестованості

1.3 Проектування тестопридатних систолічних матриць ПЕ

Тестування СМКЕ передбачає з'ясування - справна чи ні матриця загалом. В свою чергу тестування СМПЕ передбачає локалізацію несправних комірок для подальшої реконфігурації структури матриці. Це, безумовно, важливо і на стадії виробництва, і на стадії експлуатації матричних структур. Так, при виробництві матриць у вигляді НВІС через дефекти, що неминуче присутні в кристалі, частина процесорних елементів буде несправною. Отже, необхідно забезпечувати можливості обходу таких дефектних ПЕ, мінімізувати довжину загального шляху міжз'єднань і максимізувати розміри правильно функціонуючої матриці. Для цього необхідна технологія, що дозволяє розривати та створювати зв'язки між комірками (прикладом такої технології є програмування за допомогою лазера), алгоритм реконфігурації та метод локалізації несправних комірок. Що стосується стадії експлуатації матриці, локалізація несправних комірок необхідна для програмної реконфігурації структури матриці (якщо вона можлива) у процесі функціонування – це завдання систем функціонального діагностування. А якщо СМПЕ побудована на дискретних компонентах, також для заміни несправних компонентів.

Для діагностування СМПЕ найбільш зручним є використання вбудованих засобів тестування. Так як доступність внутрішніх вузлів матриці через зовнішні входи / виходи вкрай обмежена, транспортування тестових впливів до кожного ПЕ і спостереження реакцій на ці впливи утруднюються зі зростанням числа комірок, а також збільшується складність і вартість тестового обладнання.

Структура СМПЕ представляється у вигляді орієнтованого графа, вершинам якого відповідають комірки матриці, а дугам – інформаційні зв'язки між ними.

Структуру комірки в загальному випадку можна представити наступним чином. Кожна комірка складається з ПЕ і програмованого перемикача комірки (ППК), як показано на рис. 1.7. Перемикач має вхідні та вихідні порти. Їхнє число визначається видом матриці. Так, для одновимірної матриці необхідно два вхідних і один вихідний порт, а для двовимірної матриці - два вхідних і два вихідні порти. Управління перемикачем здійснюється по вертикальних і горизонтальних шин управління ППК. Перемикачі та інформаційні зв'язки між ними утворюють мережу з'єднань. Якщо мережа з'єднань комбінаційна (СМКЕ), її можна перевірити викладеним вище методом.

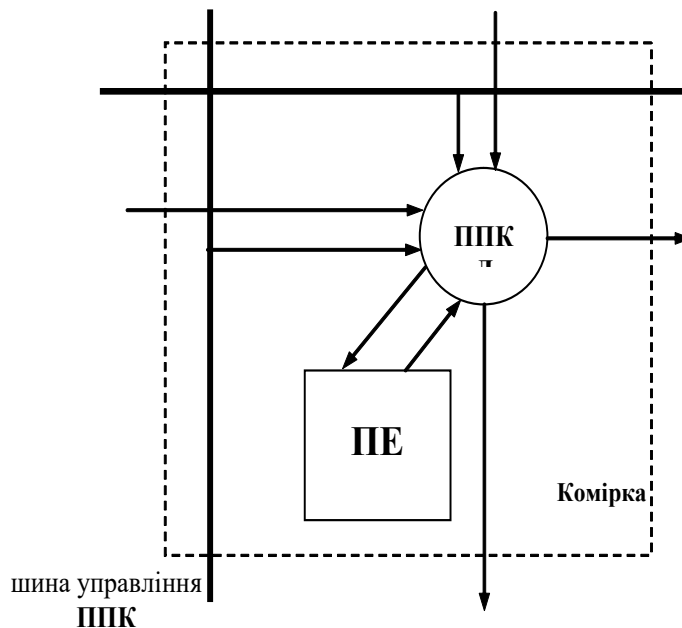


Рисунок 1.7 – Структура комірки, що реконфігурується

Найпростішим алгоритмом перевірки матриці є наступний алгоритм.

1. Засобами вбудованого тестування перевіряються одночасно всі ПЕ матриці. Результатом самоперевірки ПЕ буде служити сигнатура, причому

несправність засобів вбудованого контролю буде розглядатися як несправність ПЕ.

2. Одночасно у всіх комірках проводиться порівняння отриманих сигнатур з еталонними. У разі збігу сигнатур прапор стану комірки F встановлюється в нуль (комірка справна), інакше - в одиницю (комірка несправна).

Для перевірки даної схеми необхідний компаратор, тригер-latch для зберігання прапора стану, а також необхідне зберігання еталонної сигнатури. Для отримання останньої необхідно моделювання справної роботи ПЕ.

2 ТЕХНОЛОГІЇ РОЗВИТКУ РОЗПОДІЛЕНИХ КОМП'ЮТЕРНИХ СИСТЕМ НА ПРОГРАМОВАНИХ ФУНКЦІОНАЛЬНИХ МОДУЛЯХ

2.1 Аналіз структур та тенденцій розвитку інтерактивних комп'ютерних мереж

Широкий діапазон інтегральних мікросхем (ІМС) FPGA-технології дозволяє проектувати на їх основі широкий спектр електронних пристроїв, серед яких: засоби поєднання різних за живленням інтерфейсів, перетворювачі кодів, периферійні контролери, мікропрограмні пристрої керування, скінченні автомати, універсальні та спеціалізовані процесори, пристрої цифрової обробки сигналів тощо.

Одним з напрямів створення високопродуктивних обчислювальних систем є побудова реконфігурованих обчислювальних структур, а саме – адаптація архітектури обчислювальної системи під структуру графа задачі, в такий спосіб створюючи універсальне проблемно-орієнтоване обчислювальне середовище. Дослідження пульсуючих інформаційних решіток, як одного з прототипів однорідних структур, є актуальними під час створення реконфігурованих обчислювальних систем [1]. Розглянуто побудову реконфігурованих обчислювальних систем на обчислювальних структурах з однорідною архітектурою середовища, які мають змінність конфігурації обчислювального поля через її структурну зміну на рівні команд та послідовності самих обчислень.

В загальній структурі кристала FPGA-мікросхеми по периферії верхнього шару кристала розміщуються блоки вводу/виводу (БВВ), що можуть бути запрограмованими для виконання функцій буферів: вхідного, вихідного, із запам'ятовуванням стану та ін. У деяких серіях FPGA-ІМС рівень напруги на двох БВВ може відрізнятись, що дає змогу поєднувати різні за рівнем живлення інтерфейси. У центрі кристала у вигляді матриці

розміщено конфігуровані логічні блоки (КЛБ). Швидкодія мікросхем визначається часовою затримкою “вхід-вихід” одного КЛБ.

Область між конфігурованими логічними блоками називається областю програмованих міжз'єднань і являє собою розвинену ієрархію металічних ліній зв'язку, в місцях перетину яких розміщено спеціальні швидкодіючі транзистори. Функція області міжз'єднань полягає в забезпеченні зв'язку між будь-якими виводами КЛБ та БВВ. Необхідний маршрут міжблокових з'єднань у FPGA-ПЛІС реалізується комутацією відповідних ліній за допомогою транзисторів. Нижній шар кристала займає тіньовий запам'ятовуючий пристрій, інформація в елементах якого і визначає логічні функції КЛБ, конфігурацію БВВ та маршрути міжз'єднань.

Просторова розподіленість КЛБ FPGA; паралелізм їх функціонування та можливість програмної реконфігурації сполук та налаштувань КЛБ визначили інтерес дослідників до реалізації КС у вигляді ІКМ [2-3].

Математики використовували поняття «ітеративні комп'ютерні мережі» (ІКМ), досліджуючи ітераційні перетворення просторово-розподілених структур із дискретною множиною станів. За походженням поняття «ітеративні комп'ютерні мережі» близькі до «мереж клітинних автоматів» і позначають їх гомогенну апаратну реалізацію. Тут і далі поняття «ітеративний» та «інтерактивний» використовуються як синоніми.

З іншого боку, розвиток математичної теорії однорідних структур, концептуальних та методологічних основ їх використання у обчислювальній техніці для паралельної обробки інформації шляхом паралельного виконання обчислювальних алгоритмів, у біоінформаційних технологіях, у галузі математичного моделювання складних процесів у поєднанні із сучасними досягненнями субмікронних технологій в електронній промисловості актуальність та перспективи побудови МПСОД та її діагностичної інфраструктури на ІКМ [4-5].

ІКМ є впорядкованим масивом однорідних функціональних модулів у n -мірному просторі, в якому кожен модуль або клітина має обмежену

множину станів, а перехід з одного стану в інший визначається набором правил або функцією переходів ФМ, відповідно до якої будь-який модуль мережі обчислює своє нове стан на кожному такті функціонування мережі. Спектр застосування ІКМ надзвичайно широкий.

ІКМ є дискретними динамічними системами, поведінка яких повністю визначається правилами функціонування модулів залежно від правил взаємодій із найближчими сусідами. Значною мірою самі відносини справедливі для великого класу безперервних динамічних систем, визначених рівняннями в приватних похідних. Спільність та гнучкість використання клітинно-функціонального підходу для побудови паралельних обчислювальних мереж, моделювання та синтезу складних динамічних систем досягаються паралельною взаємодією великої кількості змінних при обмеженні їх локальними зв'язками та одноманітними правилами еволюції. ІКМ характеризується трьома основними параметрами: геометрією зв'язків сусідніх модулів, множиною правил функціонування та алгоритмом обчислення операцій [2]. Зміна кожного з наведених вище параметрів дозволяє конструювати різні ІКМ, що відрізняються своїми властивостями.

Структура ФМ ІКМ ідентична структурі КЛБ FPGA і тому будь-яка мережа з довільними правилами функціонування та еволюцією може бути конфігуровано у структурі ПЛІС. ІКМ, всі ФМ-клітини якої налаштовані за одним правилом, називають однорідною ІКМ (ОІКМ), в іншому випадку вона називається гібридною ІКМ (ГІКМ) [6].

Наразі ІКМ популярні серед вітчизняних та закордонних дослідників, які досліджують властивості мереж. В [7, 8] представлено та проаналізовано найбільш поширену трирівневу магістральну архітектуру КС та її компоненти, а також структуру фреймів неінтерактивних та ІКМ і систем. В [9] показано як Інтерактивні моделі руху даних застосовуються в КС реального часу, та показано які існують інтерфейсні виконавчі механізми мікропроцесорних систем автоматизації виробничих процесів. Аналіз існуючих методів проектування відмовостійких систем на основі

розподілених ІКМ (РІКМ) показує, що знаходять широке застосування два підходи: 1) запровадження та використання апаратної надлишковості у вигляді резервних модулів (РМ) ФМ, де РМ замінюють несправні; 2) виключення несправних ФМ із РІКМ із деградацією її продуктивності в допустимих межах (Degradable Arrays). Проблема побудови двовимірних РІКМ, що деградують, у яких використовуються чотирипортові комутатори є NP-повною. Надзвичайно складно маршрутизувати ДІКМ одночасно по рядкам і стовпцям. Було запропоновано алгоритм реконфігурації, у якому маршрутизація справних ФМ рядками і стовпцям здійснюється обмеженого за розмірністю і площі ділянки двовимірної мережі, та велику кількість алгоритмів, що мінімізують час реконфігурації, енергетичні витрати, а також альтернативні рішення при одночасній мінімізації часу та енергетичних втрат. ІКМ, структури яких не відповідають концептуальним вимогам керованості та спостережуваності, відноситься до класу «важко» тестованих мереж. Відсутність доступу до внутрішніх модулів мережі ускладнює процедуру генерації тестових наборів, оцінки їхньої ефективності та пошуку несправного модуля. Тому виникає завдання модифікації структур ІКМ, що забезпечило б підвищення ефективності процедур тестового діагностування ОІКМ та ДІКМ, та розробки методів синтезу перевіряючих послідовностей, які дозволяють виключити трудомісткі процедури моделювання виявлених несправностей [10].

ІКМ, всі ФМ-клітини якої налаштовані за одним і тим же правилом, називають однорідною ІКМ, в іншому випадку вона називається гібридною ІКМ. Однорідна або гібридна ІКМ є адитивною, якщо правила функціонування представлені класом лінійних булевих функцій [3].

Одновимірні гібридні ІКМ широко використовуються як генератори псевдовипадкових послідовностей та сигнатурних аналізаторів, а клас ІКМ, що володіє властивостями алгебри груп, як модулі помножувачів у полях Галуа, типових перетворювачів у криптосистемах. Перевагою таких пристроїв є простота та регулярність структури, а збільшення розрядності

генераторів пов'язане лише з додаванням однотипних комірок. У таких ІКМ відсутні глобальні зворотні зв'язки, характерні для генераторів на зсувних регістрах з нелінійними зворотними зв'язками.

В останні роки двовимірні ІКМ привертають увагу дослідників у галузі проектування високопродуктивних паралельних обчислювальних систем [4].

Клітинами з гомогенною топологією є гетерогенні функціональні модулі, розмірність яких змінюється від КА з двома станами до функціонального процесорного модуля. Функціональні модулі в залежності від призначення мережі можуть виконувати різні логічні та арифметичні функції: множина булевих функцій 3-х÷6-ти змінних, програмованих на рівні КЛБ FPGA; множина суматорів, помножувачів, дільників та процесорів, що програмується на FPGA або мікропроцесорних модулів, вбудованих на кристал.

Структури ІКМ представлені на рис. 2.1 досліджувалися математиками наприкінці минулого сторіччя. Як приклад цієї демонстрації наводиться в еволюції для гри у «Життя» [2], ФМ, які є клітинним автоматом на 2 стани, структуру якої та систему правил.

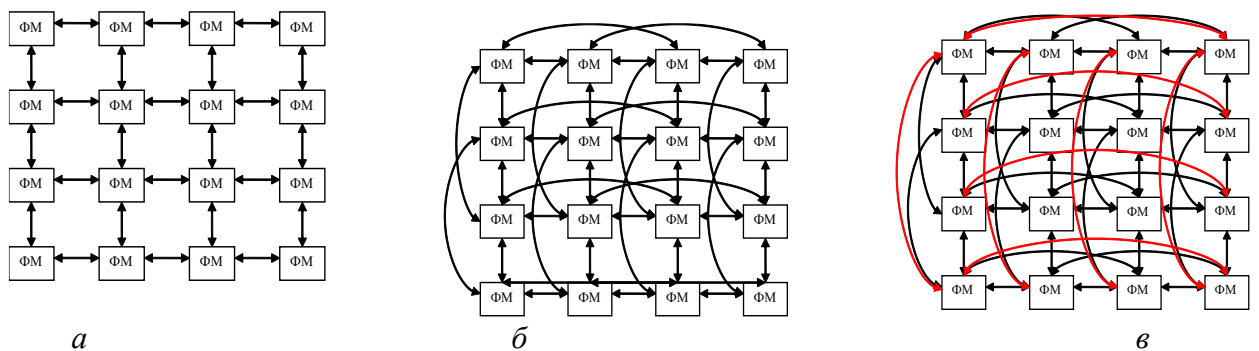


Рисунок 2.1 – Структури прямокутних ІКМ з різною топологією сполук:

- а) з'єднання ФМ та обмін даними з найближчими сусідами;
- б) з'єднання ФМ з додаванням зв'язків по рядку та стовпцю з двома сусідами;
- в) з'єднання ФМ з додаванням зв'язків по рядку та стовпцю з трьома сусідами

В інших фізичних програмах такого типу мережі клітинних автоматів (МКА) використовувалися для моделювання зростання кристалів, де було показано, що аналіз поведінки цієї мережі дозволив уточнити фізичну сутність цих складних процесів [1]. У більшості опублікованих праць для аналізу поведінки одновимірних та двовимірних ІКМ використовувався апарат алгебри багаточленів, теорії графів, формальних граматик, комбінаторної математики.

Велика кількість робіт у галузі однорідних середовищ однорідних середовищ присвячена розробці теоретичних та практичних аспектів побудови МПСОД на ІКМ, що відображають їх максимальні конструктивні та обчислювальні характеристики, можливості моделювання складних динамічних систем, реконфігурації та оптимального розміщення обчислювальних та керуючих алгоритмів, автоматних моделей у однорідному середовищі.

В інших практичних застосуваннях можливість програмної та оперативної реконфігурації ІКМ дозволяє вирішувати завдання побудови діагностичної інфраструктури та відновлення працездатності МПСОД при виявленні дефектних функціональних модулів та, таким чином, підвищити показники надійності та відмовостійкості обчислювальних та керуючих систем, що реалізуються на основі РІКМ [6].

В теперішній час ІКМ також дуже популярні серед вітчизняних та закордонних дослідників. Наприклад, такі вчені, як Я.М. Николайчук, Р.В. Цанько, Н.Я. Возна досліджують інтерактивних комп'ютерних мереж в [7, 8], де представлено та проаналізовано найбільш поширену трирівневу магістральну архітектуру комп'ютерних систем та її компоненти, а також структуру фреймів неінтерактивних та інтерактивних комп'ютерних систем. Проаналізовано програмно-апаратні засоби діагностування промислових об'єктів, методів та моделей інтерактивної взаємодії "оператор - моніторингова система". Досліджено структуру ізацію інтерактивних систем

моніторингу об'єктів. Сформульовано основні функціональні обмеження існуючих моніторингових систем, які не забезпечують оперативне виявлення та реагування операторів на складні передаварійні та аварійні ситуації на об'єктах. Обґрунтовано перспективу розробки та впровадження у моніторингових системах об'єктів структурованих образно-кластерних інтерактивних моделей "оператор - моніторингова система".

У [8] показано як інтерактивні моделі руху даних застосовуються в комп'ютерних системах реального часу, та показано які існують інтерфейсні виконавчі механізми мікропроцесорних систем автоматизації виробничих процесів. Аналіз існуючих методів проектування відмовостійких систем на основі РІКМ показує, що знаходять широке застосування два підходи:

- запровадження та використання апаратної надмірності у вигляді резервних ФМ;

- виключення несправних ФМ із РІКМ із деградацією її продуктивності в допустимих межах (Degradable Arrays).

У першому підході у РІКМ резервні модулі замінюють несправні. Відомі методи та процедури реконфігурації мереж з використанням резервних модулів представлені у роботах [1, 2]. Запропоновано використовувати графові ПМЧ моделі для визначення мінімального числа резервних ФМ, та запропоновано структуру комутаційної системи РІКМ, яка забезпечує ефективність перемикачів резервних та функціональних модулів. Однак недоліком запропонованого підходу є фіксованість та обмеженість розмірності РІКМ, що призводить до втрати функції реконфігурації ІКМ при певній кількості несправних ФМ.

У другому підході, що передбачає деградацію продуктивності системи, відсутні резервні ФМ, і вирішується завдання використання максимальної кількості справних модулів та виключення несправних ФМ для створення функціонуючої РІКМ меншої продуктивності. У ряді відомих робіт показано, що проблема побудови двомірних РІКМ, що деградують, у яких використовуються чотирипортові комутатори є NP-повною. Надзвичайно

складно маршрутизувати двовимірну мережу одночасно по рядкам і стовпцям [6]. У [9] було запропоновано алгоритм реконфігурації, у якому маршрутизація справних ФМ рядками і стовпцям здійснюється обмеженого за розмірністю і площі ділянки двовимірної мережі. Було запропоновано велику кількість алгоритмів, що мінімізують час реконфігурації [6], енергетичні витрати [9], а також альтернативні рішення при одночасній мінімізації часу та енергетичних витрат.

З розвитком РКС та бездротових телекомунікаційних систем, мобільної аудіо-відео, комп'ютерних мереж проблема маршрутизації між окремими вузлами мережі з мінімальними тимчасовими та енергетичними витратами стає особливо актуальною [10].

В даний час «тестопридатність» є одним з головних критеріїв, за яким оцінюється рівень проектування, і неформально визначається наступним чином: «об'єкт є тестопридатним, якщо процедури генерації тестів, оцінки їх ефективності та реалізації тестового діагностування можуть бути виконані в межах встановлених фінансових витрат, часу та значень показників, що характеризують пристосованість об'єкта до виявлення несправностей, пошуку місця несправності та реалізації тестового діагностування».

Запропоновано структурний метод підвищення ефективності діагностування ІКМ, заснований на введенні комутаторів у кожен комірці ІКМ, додаткового входу, виходу та схеми порівняння реакцій кожної комірки структури на тестові послідовності. Для класу однорідних середовищ вони реалізують у кожній комірці нормальний асинхронний автомат Мура чи Милі, визначено властивості, дозволяють проводити динамічний контроль зони переходів і виходів комірок середовища. Запропоновано використовувати коди з постійною вагою для побудови комірок мережі, що самоперевіряються.

Залежно від властивостей ІКМ розрізняють мережі, у яких довжина тестів тестування постійна і не залежить від кількості комірок мережі. Такі ІКМ називають С-тестованими мережами або С-ІКМ. Якщо довжина тесту,

що перевіряється, лінійно залежить від числа комірок мережі, то останні називають L-тестованими мережами або L-ІКМ.

Проблема тестового діагностування одновимірних ІКМ була досліджена в [1], в якій було запропоновано методи тестування несправностей класу в L-ІКМ. В [11] визначені необхідні та достатні умови L-тестованості одновимірної однорідної мережі без виходів, що спостерігаються, щодо класу одиночних несправностей комірок мережі. Як показано в [11] при доказі теореми – мережа тестується щодо класу одиночних несправностей комірок мережі, якщо у таблиці переходів комірки множина наступних станів містить усі стани і автоматна модель комірки є мінімальним автоматом Мура, тобто, у таблиці переходів немає двох однакових рядків. Для знаходження перевіряючих тестів мережі було запропоновано підхід, заснований на побудові тестового графа та виділення множини фундаментальних циклів у тестовому графі мережі.

2.2 Синтез перевіряючої послідовності за тестовим графом функціональної комірки ІКМ

Розроблено методи синтезу перевіряючих тестів ІКС, функціональні модулі яких не мають виходів, що спостерігаються, а функціонування представляється моделями кінцевих детермінованих автоматів. Методи синтезу перевіряючих послідовностей та побудови діагностичних експериментів, запропоновані в цій роботі, засновані на концепціях керованості та спостерігальності, які широко використовуються для оцінок тістопридатності об'єкта діагностування (ОД) багатьма дослідниками.

У деяких структурах ІКМ відсутні верхні керовані входи, що знижує показники керованості мережі, а отже ускладнює процедуру тестового діагностування (ТД) справності мережі.

З метою покращення показників керованості та спостережуваності, спрощення процедур синтезу перевіряючих тестів та діагностування

одновимірної ІКМ пропонується модифікувати структуру ІКМ та процедуру перевірки її справності відповідно до пропозицій та умов: 1) ввести до кожної комірки мережі додаткові входи x_i , $i=1,n$ для забезпечення умов керованості та транспортування несправностей у мережі на бічні виходи, що спостерігаються; 2) ввести до кожної комірки мережі додаткові виходи x_i , $i=1,n$ для забезпечення умов спостереження станів; 3) процедури перевірки справності елементів пам'яті та комбінаційної частини комірок мережі проводити окремо; 4) перевіряючий експеримент виконувати у 2 етапи відповідно до 2 напрямів поширення сигналів: ліворуч - праворуч та у зворотному напрямку, що дозволить виключити вплив зворотних зв'язків між комітками мережі та спростити процедури синтезу тестів та перевірки справності ІКМ.

При реалізації мережі на базі ПЛІС типу FPGA перелічені вище умови легко виконуються шляхом відповідного вибору типу ПЛІС та налаштування логічних блоків, що конфігуруються. Таким чином, завдання ТД ІКМ зводиться до завдання ТД одновимірної ІКМ (ОІКМ) або тільки з керованими входами в кожній комірці, або до перевірки справності ІКМ з керованими входами та спостережуваності виходів у кожній комірці мережі.

Представлений метод синтезу перевіряючих послідовностей для ІКМ без спостережуваних виходів у кожній комірці мережі. На функціональному рівні опису комірки ОІКМ з бічними виходами, що спостерігаються, будемо розглядати його таблицю істинності як таблицю переходів-виходів (ТПВ) автомата Мура, що задається трійкою (X, Z, Q) .

Така мережа є керованою, якщо в таблиці переходів комірки мережі множина станів наступників дорівнює повній множині станів. З умови керованості ІКМ випливає, що завжди існує перекладна послідовність $T(z_i, z_j)$, яка забезпечує появу на правому виході комірки $C(\alpha), \alpha = \overline{1, p-1}$, стану $z_j, j=1,n$.

Під спостережуваністю розумітимемо таку властивість ІКМ, яка забезпечує можливість ідентифікації в будь-якому внутрішньої комірки

мережі кожного стану $z_i \in Z_\alpha$ та транспортування несправності типу $z_i \rightarrow Z/z_i$ на виходи мережі, що спостерігаються. Необхідною умовою спостережуваності ІКМ є властивість мінімальності автоматної моделі комірки мережі, яка забезпечує помітність кожного стану комірки на виходах, що спостерігаються. Транспортування несправності на виходи мережі, що спостерігаються, здійснюється додатком такого вхідного вектора X_T , який дозволяє відрізнити деякий стан z_i , $i=1,n$, комірки мережі, що перевіряється, від множини інших станів Z/z_i . Складність побудови перевіряючого експерименту для ІКМ, що мають тільки бічні виходи, що спостерігаються, залежить від того, існує один або деяка множина вхідних векторів. X_T , які дозволяють відрізнити стан z_i від множини всіх інших станів комірки, що перевіряється.

Наведемо метод синтезу перевіряючих послідовностей для ІКМ без виходів, що спостерігаються. x'_i , заснований на аналізі автоматної моделі комірки мережі, побудові тестового графа (ТГ) пар станів за таблицею переходів автоматної моделі комірки та знаходження множини фундаментальних циклів у ТГ, що визначає, множина циклічних перевіряючих послідовностей усієї ІКМ. В основі побудови ТГ та його використання для знаходження безлічі перевіряючих тестів лежить поняття розрізнення пар станів, що входять до цього замкнутого циклу. Якщо у ТГ існує шлях із певної вершини $V_i = (z_a, z_b)$ до вершини V_j , що входить у деякий замкнутий цикл, то пара станів (z_a, z_b) є також помітною.

Метод синтезу перевіряючих тестів для ОІКМ, заснований на використанні ТГ комірку мережі, можна розглянути на прикладі одновимірної мережі без виходів, що спостерігаються (рис. 2.2), де мережа

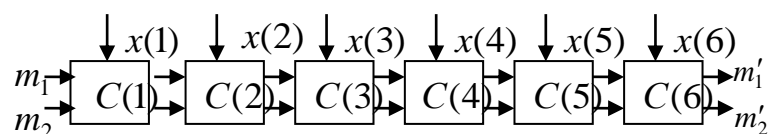


Рисунок 2.2 – Одновимірна ІКМ без виходів, що спостерігаються

складається з шести комірок із чотирма станами та однаковими правилами налаштування.

ТГ для аналізованої комірки мережі, побудований з його ТПВ і двох зв'язкових компонентів, у кожному з яких можна виділити множина фундаментальних циклів, тобто таких циклів, у яких є, щонайменше, одна дуга, що не належить ніякому іншому циклу. Якщо ТГ містить фундаментальний цикл, для виявлення всіх несправностей типу $(z_i \rightarrow z_j)$, $(z_j \rightarrow z_i)$, $(z_a \rightarrow z_e)$ і $(z_e \rightarrow z_a)$ у будь-якої комірки мережі достатньо докласти наступні чотири тести:

$$1) z_i^{x_\alpha} z_a^{x_\beta} z_i^{x_\alpha} z_a^{x_\beta} \dots 2) z_a^{x_\beta} z_i^{x_\alpha} z_a^{x_\beta} z_i^{x_\alpha} \dots 3) z_j^{x_\alpha} z_e^{x_\beta} z_j^{x_\alpha} z_e^{x_\beta} \dots 4) z_e^{x_\beta} z_j^{x_\alpha} z_e^{x_\beta} z_j^{x_\alpha} \dots$$

Незалежно від числа комірок мережі ці тести викликають появу різних станів в одній і тей ж комірці, а, отже, перелічені вище несправності виявляються на виході, що спостерігається з мережі. Оскільки множина переходів ТГ розглянутої вище мережі покривається множиною фундаментальних циклів, крім переходу $(z_0 z_1)^{x_0} z_1$, то для кожного циклу можна визначити множина тестів, що перевіряють правильність переходів та станів кожної комірки мережі. Для кожного фундаментального циклу G1-G7 ТГ комірки мережі наведено тести, що дозволяють виявити несправну комірку мережі. Однак у ТГ є перехід $(z_0 z_1)^{x_0} z_1$, який не входить до жодного з розглянутих вище циклів і породжує невизначеність під час перевірки стану комірки мережі тестом $t_7 : z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1$. Для виключення цієї невизначеності тест t_7 необхідно розширити тестами $t_{23} \div t_{27}$ ($t_{23} : z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_1} z_3$, $t_{27} : z_1^{x_0} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2^{x_1} z_1^{x_1} z_3$).

Методика синтезу перевіряючих тестів заснована на використанні ТГ комірки мережі та виділення множини фундаментальних циклів у ТГ, що має наступні особливості.

1. Складність процедури синтезу визначається складністю побудови ТГ та процедури знаходження фундаментальних циклів у графі.

2. Тести, побудовані за фундаментальними циклами ТГ комірки мережі, є, як правило, надлишковими.

3. Отримані тести можуть містити множину невизначених переходів, поява яких обумовлена наявністю в ТПВ комірки пар сумісних станів. Тому отримані тести необхідно аналізувати з метою знаходження таких невизначених переходів та подальшого розширення множини тестів, що перевіряють. Процедура аналізу та розширення множини тестів у розглянутій методиці не формалізована.

Розглянуті вище недоліки існуючої методики синтезу перевіряючих тестів для ІКМ без виходів, що спостерігаються, можна виключити, якщо скористатися підходом, заснованим на використанні характеристичних послідовностей автоматної моделі комірки та побудові перевіряючих тестів на основі цих послідовностей, що буде предметом дослідження в наступних роботах.

2.3 Концепція С-тестованості інтерактивних комп'ютерних мереж

Концепція С-тестованості ІКМ була вперше сформульована в [11]. Цей підхід був пізніше розвинений. Вперше було розглянуто методи тестового діагностування несправностей класу в ІКМ та визначено достатні умови існування лінійного тесту, що виявляє кратні несправності мережі. Проблема С-тестованості для класу несправностей одновимірної ІКМ із виходами, що спостерігаються, була досліджена, та ослаблені умови достатності виявлення кратних несправностей мережі для L-ІКМ, і отримана більш точна оцінка довжини тесту для С-ІКМ.

Проблемі тестового діагностування двовимірних ІКМ (ДІКМ) присвячено невелику кількість робіт. На жаль, достатні умови тестування ДІКМ, визначені в цих роботах, важко використовувати на практиці. Аналіз

вищезазначених робіт у галузі тестового діагностування ІКМ показує, що, незважаючи на наявні досягнення та велику кількість робіт у цій галузі, відсутній єдиний методологічний підхід до вирішення цієї проблеми для ІКМ різного класу. Так, у роботі в [11], присвяченій синтезу перевіряючих тестів в одновимірних ІКМ без виходів, що спостерігаються, по автоматній діаграмі комірки мережі будуються тестові графи, що дозволяють синтезувати множину тестових наборів. У цьому використовуються як функціональні моделі несправностей, і моделі константних несправностей.

Очевидно, що ІКМ, структури яких не відповідають концептуальним вимогам керованості та спостерігальності, відноситься до класу «важко» тестованих мереж. Відсутність доступу до внутрішніх модулів мережі ускладнює процедуру генерації тестових наборів, оцінки їхньої ефективності та пошуку несправного модуля. Тому виникає завдання модифікації структур ІКМ, що забезпечило б підвищення ефективності процедур тестового діагностування одновимірних та двовимірних ІКМ, та розробки методів синтезу перевіряючих послідовностей, які дозволяють виключити трудомісткі процедури моделювання виявлених несправностей.

2.4 Класифікація експериментів над кінцевими автоматами

2.4.1 Послідовності автоматів та їх характеристики під час проведення експерименту

Відрізняючі послідовності автоматів. Якщо відома автоматна модель ОД, то завдання побудови перевіряючого експерименту зводиться до знаходження вхід-вихідних послідовностей, які дозволяють однозначно ідентифікувати автоматну діаграму ОД. Запропонований ним підхід дає добрі результати для автоматів, які мають відмінні послідовності. Тому цей клас автоматів багато дослідників називають легкотестованим. Відома процедура знаходження відмінної послідовності автомата за його автоматною

діаграмою (графом переходів), що передбачає побудову дерева наступників автомата і застосування певних правил усікання вершин цього дерева. Процедура завершується, коли або знайдена відмінна послідовність для заданого автомата, або в результаті побудови відмінного дерева встановлено, що автомат не має відмінної послідовності. Верхня межа рангу відмінного дерева, побудованого для автомата з n станами і початковою σ -множиною потужності m має вигляд

$$h \leq (m - 1) \cdot n^m. \quad (2.1)$$

Із оцінки (2.1) і процедури побудови відмінного дерева випливає, що завдання знаходження відмінної послідовності автомата і визначення складності побудови перевіряючих експериментів, є універсальним переборним (NP-повним) завданням.

Відомо, що необхідною умовою існування для даного автомата відмінної послідовності є властивість мінімальності автомата. Однак ця умова не є достатньою, бо існують мінімальні автомати, які не мають відмінних послідовностей.

Вхідна послідовність x_0 називається відмінною для автомата $A = (X, Y, Z, \delta, \lambda)$, якщо вихідна послідовність автомата, як реакція на x_0 , різна для будь-якого початкового стану, тобто $\lambda(z_i, x_0) \neq \lambda(z_j, x_0)$, $\forall z_i, z_j \in Z, z_i \neq z_j$.

Установчі послідовності автоматів визначають, що у першій фазі діагностичного експерименту з автоматом передбачається установка його в відомий початковий стан. Це досягається використанням настановних або синхронізуючих послідовностей. Методи побудови умовних і безумовних настановних експериментів, а також синтез настановних і синхронізуючих послідовностей, досить повно і глибоко викладено, і отримана оцінка складності безумовного установчого експерименту.

Вхідна послідовність x_0 називається установчою для автомата $\langle X, Y,$

$Z, \delta, \lambda >$, якщо його кінцевий стан $\delta(z_i, X_u)$ може бути однозначно визначено по вихідній послідовності $\lambda(z_i, X_u)$ для всіх $z_i \in Z$.

Відповідно до визначення, якщо автомат справний і має установчу послідовність, то незалежно від початкового стану його можна перевести в певний стан. Будь який мінімальний автомат має послідовність. Правила усічення дерева-наступників автомата дозволяють побудувати установче дерево за його таблицею переходів-виходів, з якого визначається множина настановних послідовностей автомата. Слід зазначити, що кожна відмінна послідовність є установчою, у той час як зворотне невірно.

Для кожного мінімального автомата A з числом станів n існує безумовна установча послідовність, довжина якої не перевищує $(n - 1)^2$.

Оцінка довжини установчого експерименту є найбільшою верхньою межею, для класу мінімальних автоматів із n станами можна побудувати інсталяційний експеримент, довжина якого не перевищує $n(n - 1) / 2$, що є найменшою верхньою межею довжини установчої послідовності.

Якщо автомат мінімальний, то завжди можна знайти умовну установчу послідовність, за допомогою якої визначається кінцевий стан автомата. Для того, щоб перша фаза експерименту була повністю безумовною, автомат повинен мати синхронізуючу послідовність.

Далі визначені методи синтезу синхронізуючих послідовностей (СП) за таблицею переходів-виходів автомата, і за функціональною схемою ОД визначена верхня межа довжини СП, яка порівнюється з відомими оцінками, аналізується складність побудови СП і умови існування в автоматі однорідних синхронізуючих послідовностей.

Вхідна послідовність X_s автомата, яка встановлює його в певний кінцевий стан незалежно від стану виходу і початкового стану, називається синхронізуючою послідовністю.

Якщо автомат $A = \langle X, Y, Z, \delta, \lambda \rangle$ заданий таблицею переходів-виходів, то з визначення випливає, що автомат має синхронізуючу послідовність тоді і тільки тоді, коли існує вхідна послідовність X_s така, що

$\delta(z_i, X_u) = z_0, \forall z_i \in Z, z_0 \in Z$. множина переходів $\delta(z_i, X_s) = z_0, \forall z_i \in Z$ автомата визначає відображення множини його станів Z у якийсь певний стан z_0 при подачі на автомат вхідної послідовності X_s , тобто $z \xrightarrow{X_s} z_0$.

Синхронізуюча послідовність для заданого автомата може бути знайдена з синхронізуючого дерева, яке є деревом наступників, побудованим за певними правилами.

Так як синхронізуюча послідовність не пов'язана з аналізом вихідної послідовності автомата, то функції виходів автомата не розглядаються при побудові синхронізуючого дерева. Вершини синхронізуючого дерева відзначаються σ - множинами, які є x_i - наступниками ($\forall x_i \in X$) станів входять до σ -множини попередніх вершин.

Довжина синхронізуючої послідовності не перевищує суми виду:

$$C_n^{n-1} + C_n^{n-2} + \dots + C_n^2 = \sum_{k=2}^{n-1} C_n^k.$$

Через те, що $2^n = \sum_{k=0}^n C_n^k = 1+n+\sum_{k=2}^{n-1} C_n^k$ то, відповідно, найдовший шлях в синхронізуючому дереві, який може завершитися відповідно до правила I, складається з $2^m(n+1)$ вхідних символів.

Найкоротша синхронізуюча послідовність може мати довжину в один вхідний символ.

На практиці всі ОД, елементи пам'яті яких мають сигнал скидання, можуть встановлюватися в початковий стан синхронізуючою послідовністю мінімальної довжини.

Необхідні умови, яким повинен задовольняти автомат, що володіє синхронізуючою послідовністю, визначаються наступною умовою.

Автомат має синхронізуючу послідовність, якщо існує принаймні один вхідний символ $x_k \in X$ такий, що $\delta(z_i, x_k) = \delta(z_j, x_k)$, де $i \neq j, 1 \leq i, j \leq n, n \geq 2$.

Характеристичні послідовності (ХП) є важливим класом вхідних

послідовностей, якими володіє будь-який мінімальний кінцевий автомат. Мінімальний автомат з n станами має не більше $(n - 1)$ ХП і довжина кожної послідовності не перевищує $(n - 1)$ символів.

Характеристичне дерево є дерево-наступників, кожна вершина якого відзначається σ -множинами помітних станів-наступників і породжуючих їх станів попередників початкової σ - множини, які утворюють π - розбиття початкового σ - множини станів автомата на підмножини помітних станів.

2.4.2 Методи підвищення тестопридатності кінцевих автоматів

Загальні витрати, пов'язані з діагностичним забезпеченням пристрою на етапі життєвого циклу, вважаються в даний час, як правило, надмірно великими і, ймовірно, залишатимуться такими для багатьох сфер електронної промисловості від виробництва мікросхем до виробництва закінченої системи. У результаті було докладено значних зусиль, спрямованих на скорочення окремих видів витрат. Це зумовило розвиток методів проектування тестопридатності пристроїв і появу надійних і життєздатних методів проектування.

Схема є тестопридатною, якщо процедури генерації множини тестових наборів, оцінки їх ефективності та реалізації тестового діагностування можуть бути виконані за умови дотримання у встановлених межах фінансових витрат, витрат часу і значень показників, що характеризують пристосованість схеми до виявлення несправностей, пошуку місця несправностей і реалізації тестового діагностування.

Ключовим моментом цього визначення є необхідність виконання певних технічних вимог у межах допустимих фінансових витрат. Якщо реальна вартість тестового діагностування більше встановленої, то відповідно до наведеного визначення виріб спочатку було спроектовано нетестопридатним. Зробити виріб тестопридатним можна або шляхом збільшення допустимих витрат, або шляхом скорочення вартості одного або

декількох головних чинників, що визначають вартість програмного тестування. Якщо прийнятий другий підхід, то будь-яка процедура, яка забезпечує зменшення витрат на тестове діагностування, може розглядатися як процедура тестопридатного проектування.

Розглядаються три основні підходи до скорочення витрат на тестове діагностування шляхом проектування тестопридатності логічних схем. Підходи можна класифікувати наступним чином:

а) чисельна оцінка характеристик спостережуваності та управління схеми, що дозволяє оцінити міру тестопридатності схеми, яка може використовуватися на етапі проектування пристроїв;

б) методи структурного проектування тестопридатності схем і в деяких випадках самотестованих схем, засновані на використанні властивостей об'єкту сканування шляху, що забезпечує простоту доступу до внутрішніх точок схеми, а також ефективні методи генерації тестів для схем, що реалізують методи сканування;

в) розробка переліку практичних керівних вказівок, які забезпечують скорочення витрат на процедури генерації тестів і реалізації тестового діагностування.

Переваги використання тестопридатного проектування: можна узагальнити і представити однією фразою – скорочення часу повного циклу проектування і вартості тестового діагностування без зниження якості виробу. Крім того, окремі етапи процесу тестового діагностування стають контрольованими, а розробники обізнаними в методах тестового діагностування. Тільки таким шляхом можна дійсно об'єднати процеси проектування і тестового діагностування. Якщо розглядати як об'єкт діагностування HDL-моделі кінцевих автоматів у формі автоматного шаблону, то для побудови на їх основі легкотестованих автоматів розглянуті моделі повинні відповідати таким умовам:

– у моделі повинен бути спеціальний режим переводу автомата в режим тестування і навпаки на будь-якому такті роботи автомата;

– у режим тестування автомат може бути встановлений у будь-який стан за $(n-1)$ тактів, де n - число станів автомата і може бути організований гамільтонов цикл для будь-якого зі станів автомата;

– легкотестований автомат повинен будуватися в автоматизованому режимі засобами САПР.

В області логічного проектування існує множина пропозицій, спрямованих на створення тестопридатності пристроїв. Багато методів, заснованих на цих пропозиціях, або застосовуються лише до проектування комбінаційних схем, або мають істотні недоліки, обумовлені необхідністю введення додаткових первинних входів, виходів і вентилів. Проте в цьому розділі описуються деякі з цих методів, які протягом тривалого часу проявили свою «життєздатність» на практиці логічного проектування. Основний принцип методів структурного тестопридатного проектування схем полягає в організації сканування вхідних і вихідних даних або просто «сканування».

Це служить підставою для обговорення в цій роботі методів сканування, що застосовуються при проектуванні логічних пристроїв, і в цьому розділі представлено опис трьох варіантів реалізації методів сканування: сканування шляху (Scan Path), сканування з довільним доступом (Random Access Scan), сканування, чутливого до рівня тактового сигналу (LSSD - Level Sensitive Scan Design). Кожен із цих методів розроблявся великими компаніями, зокрема, метод LSSD, розроблений фірмою IBM, привернув увагу багатьох фахівців.

Проектування з використанням методів сканування слід розглядати як ретельно продуману спробу спростити процедуру генерації тестів для логічних пристроїв, що містять елементи пам'яті і ланцюги глобального зворотного зв'язку.

Метод сканованого шляху дозволяє вирішувати цю проблему шляхом зменшення складності структури схеми. Принцип, що лежить в основі метода, полягає в наступному:

- елементи пам'яті перевіряються окремо від усієї схеми та спочатку;
- Комбінаційна частина (КЧ) перевіряється в другу чергу з можливістю установки внутрішніх змінних у будь-який стан і спостережуваності виходів КЧ безпосередньо.

Це можливо, якщо в пристрої реалізувати метод сканованого шляху через елементи пам'яті (рис. 2.3). Дійсно, кожному елементу пам'яті передусє мультимплексор «2-1», керований загальним сигналом вибору режиму сканування. Коли цей сигнал дорівнює 0, мультимплексори з'єднують виходи комбінаційної частини схеми зі входами елементів пам'яті, тощо. Схема працює в режимі нормального функціонування. Коли сигнал управління сканування дорівнює 1, елементи пам'яті реконфігуруються в один зсувний регістр з послідовним введенням і виведенням даних. Вхід, на який дані надходять послідовно, називають входом сканованих даних, а вихід, з якого послідовно зчитуються дані, називають виходом сканованих даних. У режимі сканування елементи пам'яті можна просто встановити в будь-який заданий із множини станів, подаючи послідовність сигналів на вхід сканованих даних і тактуючи зсувний регістр за допомогою системного тактового імпульсу.

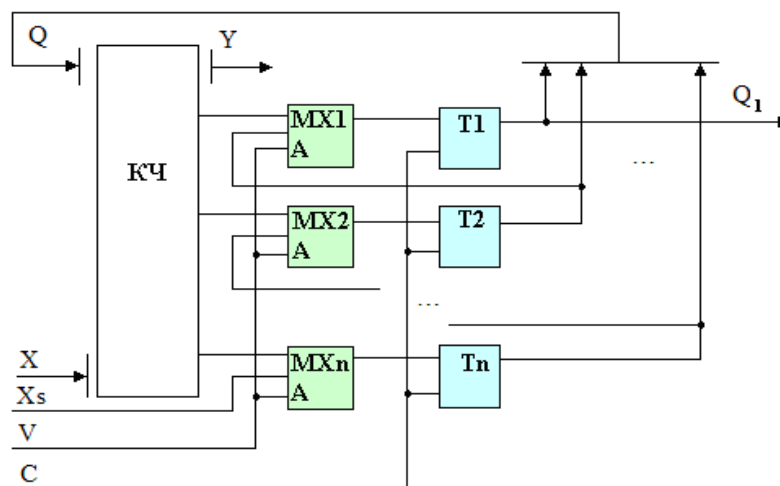


Рисунок 2.3 – Принцип сканованого шляху

Схема складається з комбінаційної частини (КЧ) і послідовної частини (ПЧ) (набору тригерів Т), де: X – зовнішні (первинні) входи; Y –

зовнішні (первинні) виходи; Q – внутрішні змінні; C – вхід синхронізації.

Кожному елементу пам'яті передують мультиплексор "2 в 1", керований загальним сигналом вибору режиму сканування V. Коли $V = 0$, вибирається режим нормального функціонування (F) – мультиплексори з'єднують виходи КЧ із входами тригерів. Коли $V = 1$, обирається режим сканування шляху (SP) – елементи пам'яті реконфігуруються в один зсувний регістр з послідовним введенням - виведенням даних. X_s - вхід сканованих даних, на нього дані надходять послідовно біт за бітом. Q1 - вихід сканованих даних, з нього послідовно зчитуються зрушувані дані.

Таким чином, в режимі SP елементи пам'яті можна досить просто встановити в будь-який заданий стан, подаючи послідовність сигналів на X_s , тактуючи зсувний регістр за допомогою синхроімпульса C.

Послідовність тестування при цьому наступна.

1. Встановити режим сканування, тобто реконфігурувати елементи пам'яті в зсувний регістр. Перевірити стани і справну роботу пристроїв кожного елемента пам'яті, шляхом використання входу-виходу сканованих даних і системного тактового імпульсу за допомогою тестів.

2. Визначити множину тестів для комбінаційної частини схеми, припускаючи повне управління всіма входами (первинними і внутрішніми входами елементів пам'яті); безпосереднє спостереження станів всіх виходів (первинних виходів і виходів елементів пам'яті).

3. Застосувати кожен тест наступним чином: а) встановити режим сканування шляху, в елементи пам'яті попередньо записати тестові набори і встановити додатково тестові дані на зовнішніх входах; б) встановити режим нормального функціонування, стани внутрішніх виходів комбінаційної частини записати в елементи пам'яті, синхронізуючи запис системним тактовим імпульсом; в) повернутися до режиму сканування і синхронно з тактовим імпульсом вивести вміст елементів пам'яті через вихід сканованих даних, порівняти цю послідовність і спостережувані стани зовнішніх виходів схеми з еталонною реакцією справної схеми.

3 МЕТОДИ ПРОЕКТУВАННЯ РЕКОНФІГУРОВАНИХ ІКМ НА ПРОГРАМОВАНИХ ФУНКЦІОНАЛЬНИХ МОДУЛЯХ

3.1 Синтез одновимірних ІКМ із централізованим управлінням реконфігурацією

Аналіз структурної організації ІКМ, наведений у розділі 1.1, показує, що розмірність клітинних функціональних модулів може змінюватися в широких межах залежно від оброблюваних мережевих інформаційних потоків, класу завдань і обчислювальних алгоритмів. Часто такі мережі належать до класу гібридних ІКМ, у якому кожна ФМ використовує різні механізми перетворення інформаційного потоку вихідних даних. На рис. 3.1 представлена найбільш поширена структура ІКМ з централізованим управлінням процесом обробки даних і мережею, що реконфігурується. У цьому модулі ІКМ з'єднуються в одновимірну мережу з односпрямованими зв'язками між модулями [12, 13].

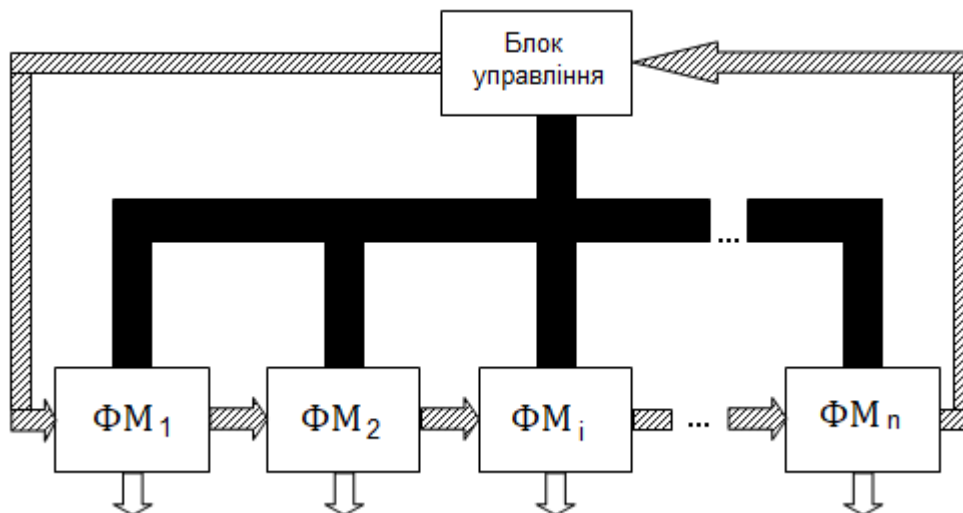


Рисунок 3.1 – Однонаправлена ІКМ із функціональних модулів

У цьому розділі представлені та розроблені методи та процедури синтезу одновимірних ІКМ, що дозволяють використовувати резервні ФМ

без зниження продуктивності мережі та засновані на вирішенні задачі мінімаксного розміщення резервних ФМ, що забезпечують реконфігурацію структури ІКМ з мінімальними часовими витратами на перекомутацію та пересилання даних між ФМ та первісної архітектури та обчислювальних характеристик ІКМ [6, 9, 14].

Запропоновані методи та процедури реконфігурації ІКМ функціональних модулів розглянуті без зниження спільності реалізації процедур за умови наявності у мережі одного несправного модуля. Цей підхід може застосовуватися до модульних процесорних мереж різної розмірності та призначення, в яких несправний ФМ виявляється або зовнішніми або вбудованими засобами діагностування.

Якщо між модулями мережі немає взаємозв'язку, то найпростішим рішенням є вихідне підключення резервних модулів до загальної шини. У разі відмови працюючого модуля включається один із резервних, виконується пересилання даних і несправний модуль вимикається.

Наявність взаємозв'язків між модулями мережі ускладнює задачу реконфігурації ІКМ. У цьому випадку явно недостатньо підключити резервний модуль та вимкнути несправний. Повинна бути створена нова схема інтерфейсних зв'язків між модулями, яка залежить від сфери застосування ІКМ. На рис 3.1 представлено найбільш поширену структуру ІКМ, яка надалі досліджуватиметься більш детально. У цій ІКМ модулі з'єднані в одновимірну мережу з односпрямованими зв'язками між модулями. Очевидно, що при відмові одного з модулів така система стає непридатною. У разі відмови модуля необхідно виконати таку послідовність операцій:

- локалізувати несправні модулі;
- визначити нову робочу конфігурацію структури;
- виконати відповідне пересилання даних;
- реконфігурувати систему шляхом перекомутації модулів.

У цьому підході управління реконфігурацією здійснює комп'ютер HOST з урахуванням:

- а) обсягу даних, які необхідно переслати перед виконанням реконфігурації;
- б) складності алгоритму реконфігурації;
- в) числа резервних модулів, необхідних для забезпечення необхідного рівня відмовостійкості системи;
- г) складності комутаційної схеми, яка потрібна на виконання реконфігурації.

Приклад структури ІКМ, де реконфігурація виконується на основі використання комутаційної мережі наведено на рис. 3.2. У цій структурі допускаються довільні з'єднання між модулями. На рис. 3.2, а представлена однорідна ІКМ з $(m + s)$ ФМ та комутаційною мережею.

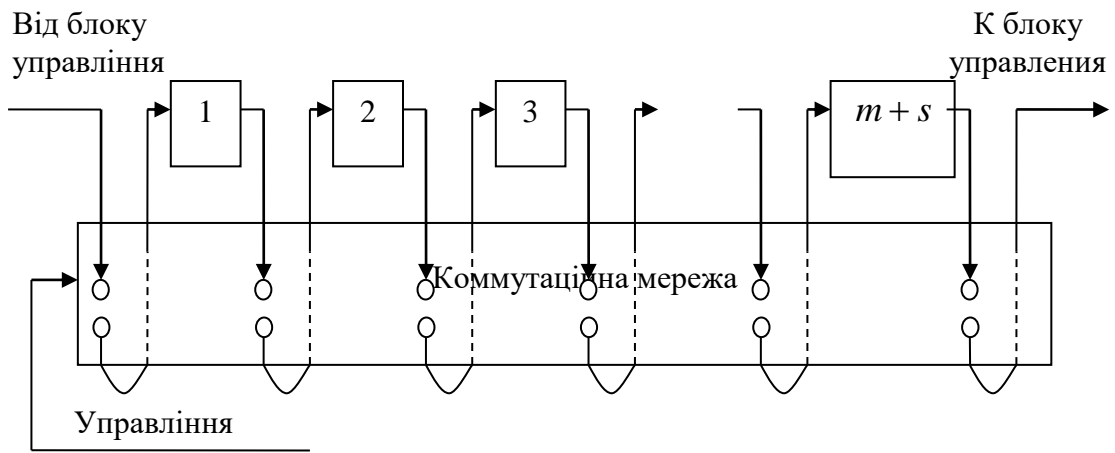
Для кожного інтерфейсного зв'язку між ФМ комутаційна мережа повинна мати вхідних і стільки ж вихідних контактів. Однак у кожен момент часу використовується тільки m з'єднань між цими контактами, у зв'язку з цим необхідно використовувати комутаційну мережу з неповною перекомутацією, яка може бути реалізована економічніше.

На рис. 3.2 (б, в) ілюструється процес реконфігурації за наявності несправного ФМ під №2 в ОС з s – число ФМ у структурі ОС та m – число резервних ФМ.

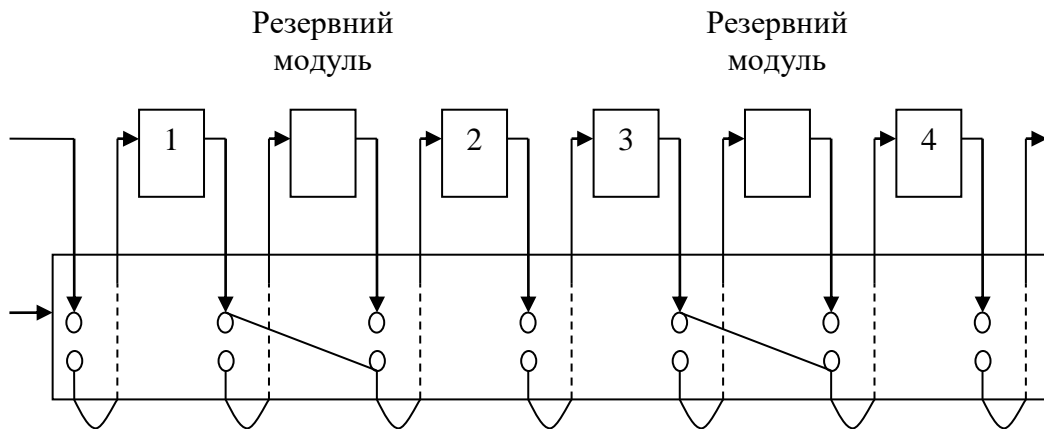
Основним недоліком даної схеми є значна складність комутаційної мережі, яка зростає пропорційно до зростання числа m ФМ та ліній зв'язку між ними. Крім того, така ІКМ є ядром обчислювальної системи, яка має бути захищена від стійких і несправностей, що перемежуються, що створює додатково проблеми при проектуванні ІКМ.

Метод використання шунтуючих комутаційних мереж був запропонований у [4]. Такі мережі реконфігуруються простими комутаційними комірками з двома типами з'єднань, що мають по два входи та виходи. Приклад такої комірки та двох можливих варіантів сполучень

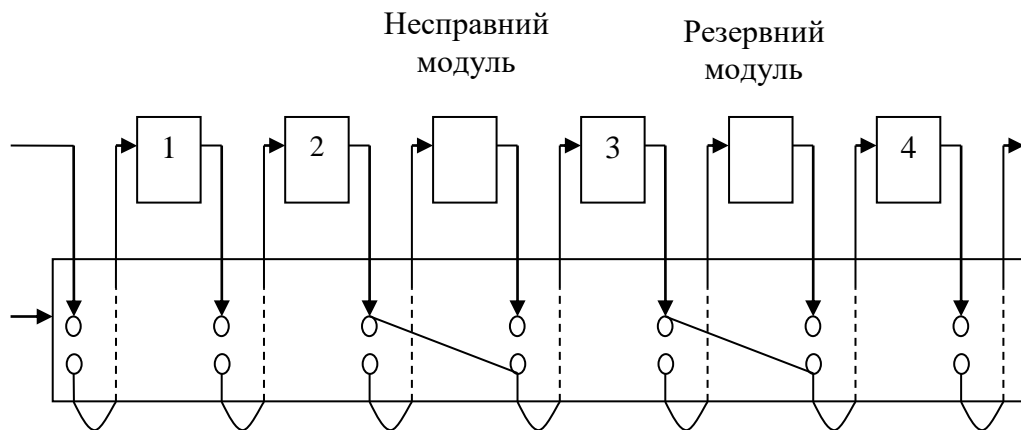
наведено на рис. 3.3. На рис. 3.4 показано використання таких комірок у механізмі реконфігурації.



а



б



в

Рисунок 3.2 – Реконфігурація ІКМ на основі комутаційної мережі: а) ФМ, з'єднані через мережу комутацій; б) структура із двома резервними модулями; в) структура мережі з несправним модулем 2

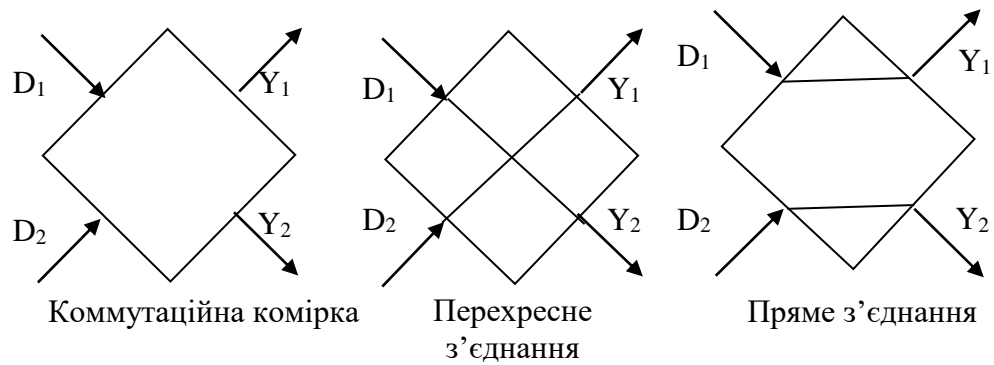


Рисунок 3.3 – Комутаційна комірка з двома типами з'єднань

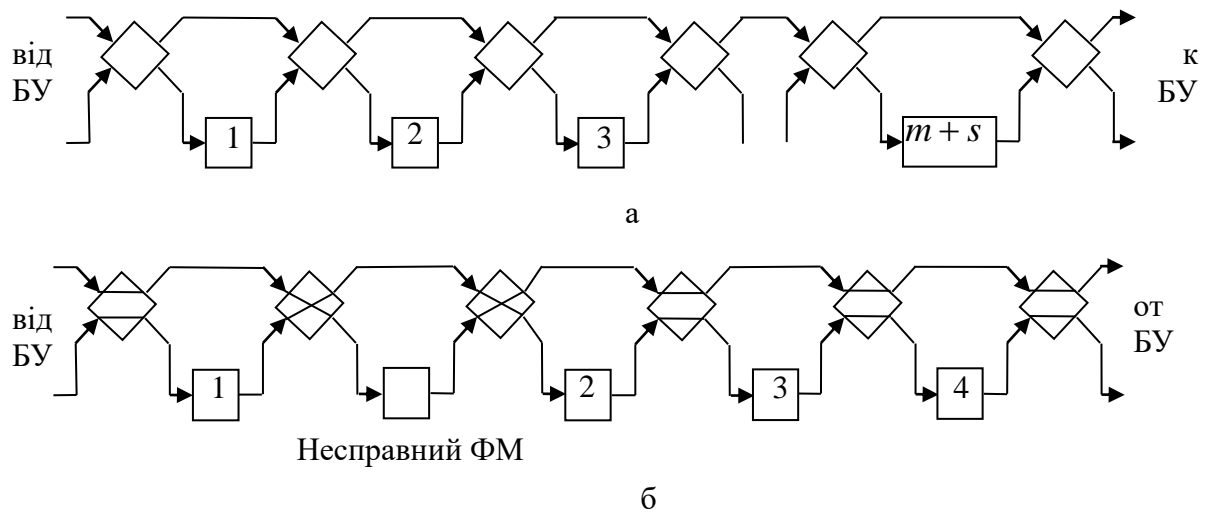


Рисунок 3.4 – Реконфігурація на основі комутаційної мережі, що шунтує:
 а) $(m + s)$ ФМ, з'єднаних через шунтуючу мережу; б) структура мережі з
 несправним ФМ 2

Як видно із рис. 3.4 недоліком даної схеми є необхідність пересилання великого обсягу даних у разі виявлення несправного ФМ. Число таких пересилок, а отже, і час реконфігурації, можна скоротити шляхом оптимального розміщення резервних модулів у структурі ІКМ. Відомо, що завдання оптимального пошуку деякого елемента множини із заданими властивостями серед множини інших елементів за мінімальне число кроків вирішується методом дихотомії або ітеративного розбиття вихідної множини на рівні частини. З цього випливає, що за наявності в мережі m працюючих та

одного резервного модуля кількість необхідних пересилань даних для модуля, що знаходиться в k -й позиції ($i \leq k \leq m+1$), є максимальним для $k=1$ або $m+1$ та мінімальним для $k = m+1/2$ для непарних m та $(m+2)/4$ для парних m .

При використанні більшої кількості резервних модулів ($s > 1$) рівняння, що визначає оптимальне розміщення, скласти набагато складніше. Пропонується наступний підхід до вирішення цього завдання. Припустимо, в ІКМ, яка має p несправних модулів ($p < s$) відмовляє $(p+1)$ -й модуль. Нехай резервні модулі знаходяться у позиціях $k_1 < k_2 < \dots < k_s$. Їхнє розташування є оптимальним лише до першої відмови, інтуїтивно можна припустити, що оптимальним розміщенням є розташування резервних модулів у середині групи з $1 + m/s$ модулів.

Визначається середня кількість пересилок при першій відмові як функція від $(k_1 < k_2 < \dots < k_s)$ змінних. Отримана функція мінімізується. Для несправного модуля з порядковим номером меншим, ніж k_1 , середня кількість пересилок складе:

$$T = \frac{1}{k_1 - 1} \cdot [(k_1 - 1) + (k_1 - 2) + \dots + 1] = \frac{k_1}{2}. \quad (3.1)$$

Для несправного модуля з порядковим номером, укладеним між k_{p+1} та k_p $p=1,2,\dots,(s-1)$, середня кількість пересилок подається у вигляді:

для непарних значень $k_{p+1} - k_p$:

$$T = \frac{k_{p+1} - k_{p-1}}{4}, \quad (3.2)$$

для парних значень $k_{p+1} - k_p$:

$$T = \frac{k_{p+1} - k_{p-1}}{4} + \frac{1}{4 \cdot (k_{p+1} - k_{p-1} - 1)}, \quad (3.3)$$

З метою спрощення можна знехтувати величиною $1/[4 \cdot (k_{p+1} - k_p)]$.

$$\begin{aligned}
 k_2 &= 3 \cdot k_1 - 1 \\
 k_3 &= 5 \cdot k_1 - 2 \\
 k_4 &= 7 \cdot k_1 - 3 \\
 &\dots\dots\dots \\
 k_s &= (2 \cdot s - 1) \cdot k_1 - (s - 1)
 \end{aligned}
 \tag{3.9}$$

Виконуючи послідовну підстановку рівнянь (3.8) та (3.9), отримуємо рівняння, що зв'язує номер розташування першого резервного ФМ у мережі з m функціональних ФМ та s резервних у вигляді:

$$\begin{aligned}
 -(2 \cdot s - 3) \cdot k_1 + (s - 2) + 3 \cdot (2 \cdot s - 1) \cdot k_1 - 3 \cdot (2 \cdot s - 1) &= \\
 &= 2 \cdot m + 2 \cdot s + 1
 \end{aligned}
 \tag{3.10}$$

або шляхом спрощення розрахункової формули (3.10) отримуємо

$$k_1 = 1 + \frac{m}{2 \cdot s}
 \tag{3.11}$$

Аналогічно, для обчислення значення k_r - номера розміщення r -го резервного модуля отримуємо загальне рівняння у вигляді:

$$k_r = r + \frac{m \cdot (2 \cdot r - 1)}{2 \cdot s}, \quad r = 1, 2, \dots, s.
 \tag{3.12}$$

Приклад 3.1. Синтезувати структуру ІКМ із $m = 16$ функціональних ФМ та $s = 3$ резервних модулів з мінімальним часом реконфігурації. З (3.12) визначаємо номери позицій резервних ФМ: $k_1 = 4$; $k_2 = 10$; $k_3 = 16$. Структура мережі, що реконфігурується, представлена на рис. 3.5.

Нижче наведено процедуру реконфігурації ІКМ із використанням шунтуючих комутаційних модулів. В алгоритмі використовуються два узагальнені характеристичні вектори: 1) вектор станів ФМ (0 – резервний, 1 – функціонуючий, 2 – несправний); 2) вектор станів комутаційної комірки (0 – перехресне з'єднання, 1 – пряме з'єднання).

До вхідних даних процедури відноситься номер несправного ФМ. У процесі обчислень змінюються значення векторів станів ФМ та комутаційних комірок, а також ініціюється процес пересилання даних. Фактично, вектор

стану комутаційної комірки можна використовувати, оскільки його значення однозначно визначається вектором станів ФМ. Однак надалі цей вектор використовується, оскільки процедура його модифікації є досить простою.

Процедура реконфігурації ІКМ може бути подана у вигляді послідовності наступних кроків.

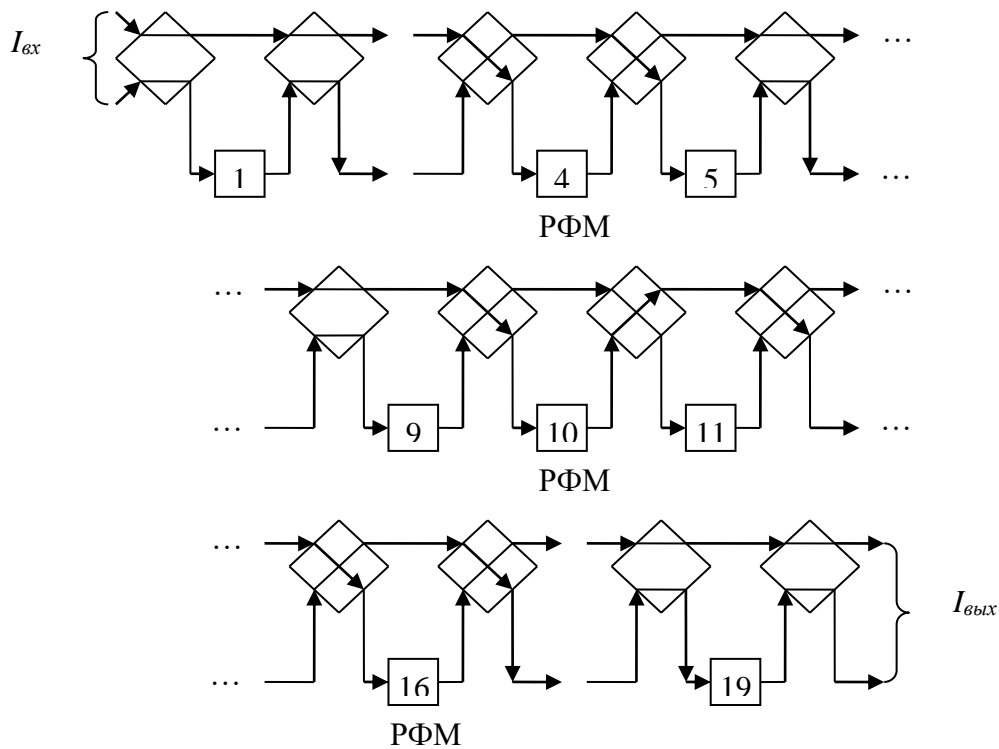


Рисунок 3.5 – Структура реконфігурованої ІКМ с резервними ФМ в 4, 10 и 16 позиціях

Алгоритм 3.1.

Крок 1. Перевірити наявність резервних ФМ у мережі. Якщо такий ФМ є, перейти до кроку алгоритму 2, в іншому випадку виконати функцію «відмова системи» і перейти до кроку 7.

Крок 2. Визначити номер позиції резервного ФМ, використання якого дозволяє досягти мінімальної кількості пересилок даних.

Крок 3. Модифікувати значення вектора станів комутаційних комірок на входах і виходах як несправного, так і резервного ФМ, що знову

включається.

Крок 4. Модифікувати значення вектора станів ФМ, враховував новий стан резервного ФМ, що підключається.

Крок 5. Виконати функцію пересилання даних між ФМ, у разі руйнування даних виконати процедуру відновлення шляхом повторного виконання фрагментів програми.

Крок 6. Модифікувати значення вектора станів ФМ мережі, враховував позицію несправного ФМ та перейти до кроку 7.

Крок 7. Кінець алгоритму.

Так як шунтуюча комутаційна мережа є частиною ядра системи, вона повинна бути також стійкою до відмови.

Працездатність шунтуючої мережі може бути відновлена при виникненні несправностей, які можуть бути виявлені за допомогою схем контролю, що самоперевіряються. У разі використання надмірної комутаційної мережі, наведеної на рис. 3.6, можна відновити працездатність мережі у разі несправних комутаційних комірок.

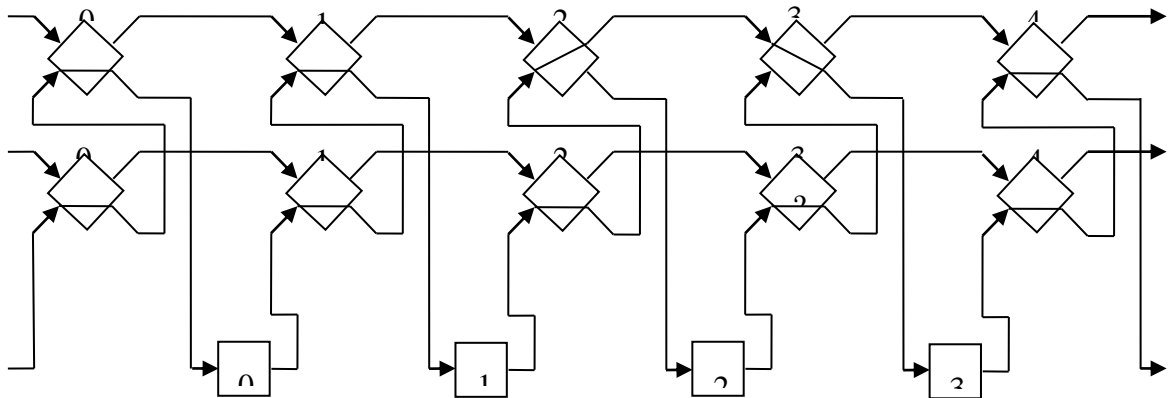


Рисунок 3.6 – Надлишкова шунтуюча комутаційна мережа

Як показано на рис. 3.6, у разі відмови комутаційні комірки 3, передача даних через мережу не переривається. Однак, внаслідок такої несправності ФМ 2 надалі не використовується.

3.2. Синтез ІКМ із розподіленим управлінням реконфігурацією

3.2.1. Одновимірні односпрямовані мережі

У мережах з розподіленим керуванням реконфігурацією комунаційні модулі, що шунтують, замінюються системою внутрішньої комутації вход-вихідних шин, вбудованою в кожен ФМ мережі [2]. У цьому випадку, кожен ФМ має дві множини входів і дві множини виходів (рис. 3.7). Такий ФМ може мати чотири маршрути передачі даних, які в залежності від входів, що використовуються, - виходів позначаються: НН, НV, VН, та VV.

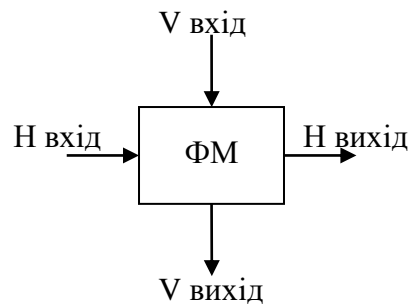


Рисунок 3.7 – ФМ ІКМ з розподіленим керуванням конфігурацією

Існує ряд альтернативних методів структурної організації розподіленої мережі, які відрізняються апаратною надмірністю, що вноситься, яка забезпечує необхідний рівень відмовостійкості ІКМ. На рис. 3.8 а представлена структура мережі з одним резервним ФМ ($s = 1$).

Рис. 3.8 б, ілюструють структури мережі з справними ФМ ($m = 5$) і при відмові модуля 3. Очевидно, що у разі відмови будь-якого ФМ мережа реконфігурується в працездатну без зниження її продуктивності.

Така структура має такі переваги:

- 1) пересилання даних здійснюється лише з одного ФМ;

2) можна поліпшити рівень відмовостійкості ОС у разі деяких кратних несправностей ФМ шляхом запровадження резервних ФМ окремих груп функціонуючих модулів.

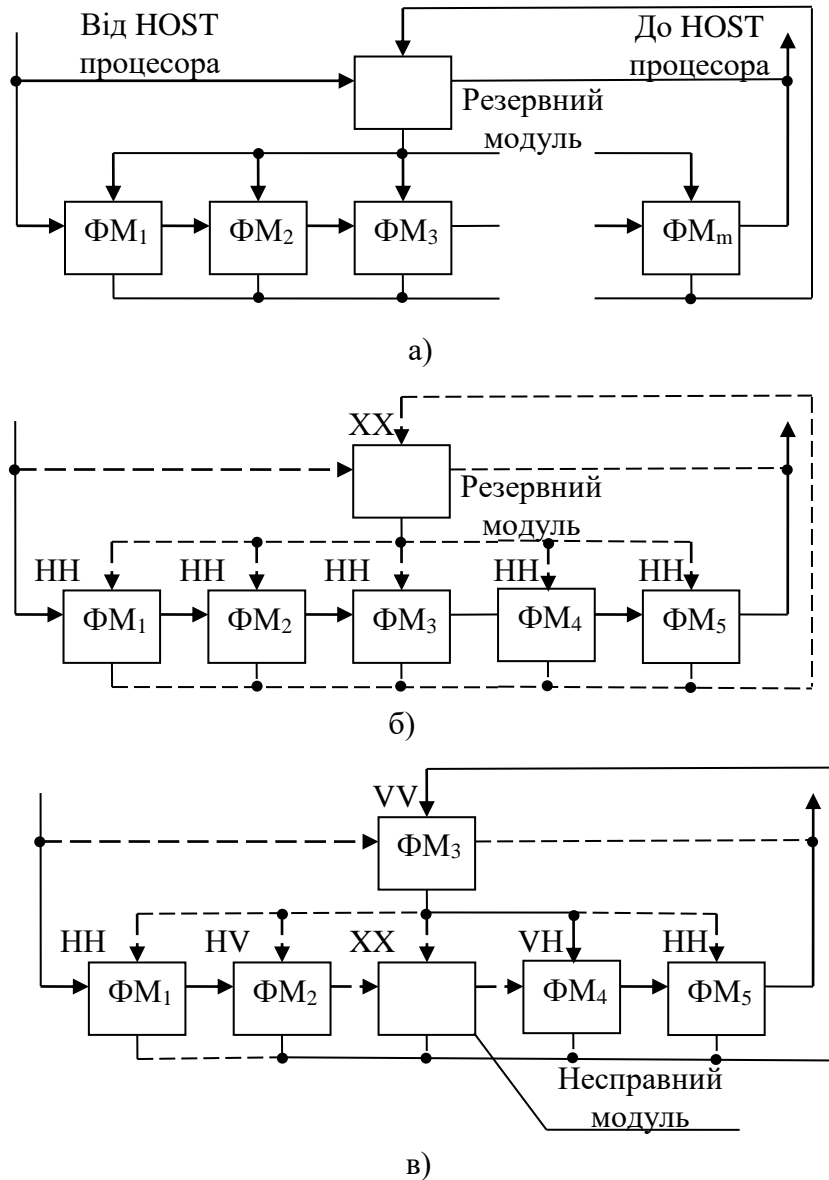


Рисунок 3.8 – Структура мережі з розподіленим управлінням реконфігурацією з одним резервним ФМ: а) структура вихідної мережі; б) структура мережі при нормальному функціонуванні з $m = 5$; в) структура мережі з несправним модулем 3

Основним недоліком такої структури мережі є можливість спотворення даних на загальній шині у разі несприятливого впливу відключеного несправного ФМ на цю шину.

3.2.2. Двовимірна мережа з розподіленим управлінням реконфігурацією

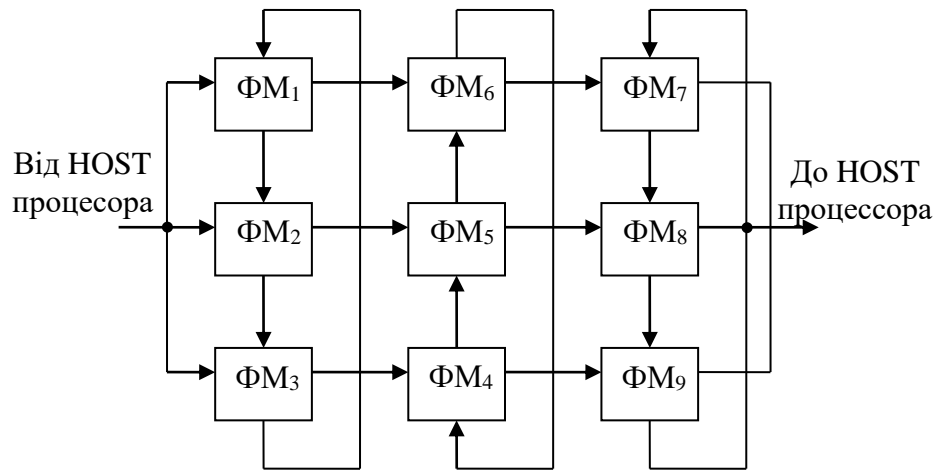
Функціональні модулі з двома вхідними та вихідними шинами можуть бути з'єднані у вигляді двовимірної мережі, як показано на рис. 3.9 а.

На відміну від структур двовимірних мереж, аналізованих у розділі 1.1 з двосторонніми зв'язками з найближчими сусідами, у мережах з обмеженою системою передачі та комутації даних у ФМ (ліворуч – праворуч, зверху – вниз і навпаки) досить просто вирішується завдання вибору оптимального маршруту обробки даних та розподіленого управління реконфігурацією.

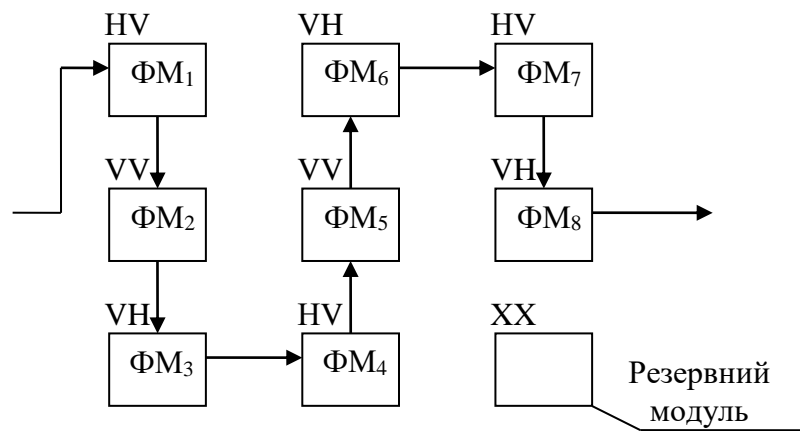
На рис. 3.9 б представлена структура мережі з параметрами ФМ. При відмові однієї з ФМ, наприклад ФМ2, мережа реконфігурується структуру рис. 3.9, шляхом заміни несправного ФМ резервним модулем. З розглянутого прикладу випливає, що для відновлення працездатності двовимірної мережі необхідно вирішити два наступні завдання:

- 1) для заданої розмірності мережі, числа резервних ФМ забезпечити її конфігурацію у разі несправних ФМ максимально можливої кратності;
- 2) синтезувати вихідну конфігурацію мережі з функціонуючих та резервних модулів для забезпечення заданого рівня стійкості до відмови.

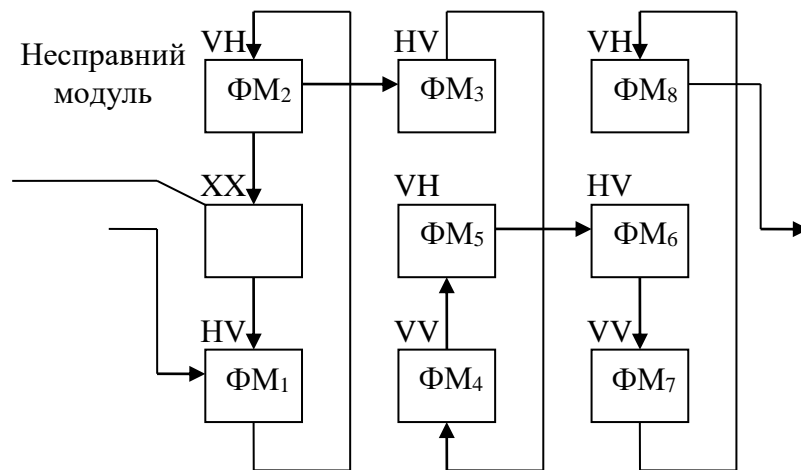
Однак досі не знайдено рішення двох зазначених вище завдань. Розглянемо двовимірну мережу з прямокутною конфігурацією та розмірністю $r \times c$ (рис. 3.10). У такій мережі з'єднання всередині рядків завжди виконуються ліворуч, а з'єднання всередині стовпців у вигляді замкнутого кільця. Слід зазначити, що з структури, зображеної на рис. 3.10, число « c » має бути парним.



а)



б)



в)

Рисунок 3.9 – Структура двовимірної мережі з розподіленим управлінням реконфігурацією: а) вихідна конфігурація мережі; б) структура мережі із одним резервним модулем; в) структура мережі з несправним модулем

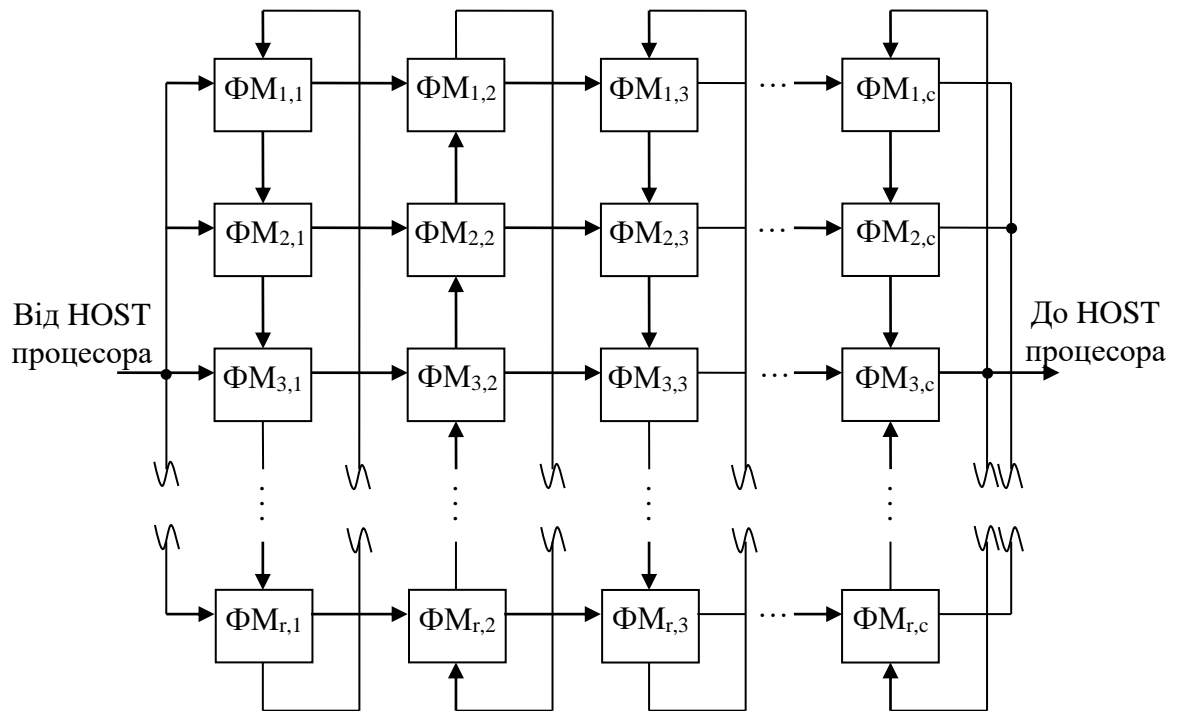


Рисунок 3.10 – Прямокутна структура мережі розмірністю $r \times c$

Правило 3.1. У мережі з прямокутною структурою розмірністю $r \times c$ число модулів у стані HV дорівнює числу модулів у стані VH і має такі межі:

$$\frac{(m-c)}{(r-1)} \leq n_{HV} = n_{VH} \leq c, \quad (3.13)$$

де m – кількість ФМ.

Робочій конфігурації відповідає ланцюжок, що складається з m ФМ, починається з вхідного і закінчується вихідним ФМ. Розглянемо i -ий стовпець. Шлях даних проходить через кілька ФМ цього стовпця, і потім переходить до наступного стовпця. Якщо в стовпці використовується лише один ФМ, то останній перебуває в стані HH і такий стовпець не впливає на величину чисел n_{HV} та n_{VH} . Якщо у стовпці i використовується кілька ФМ, перший з них перебуватиме в стані HV , останній у стані VH , а ті, що залишилися - в стані VV . Таким чином, у i -м ФМ стовпці збільшується на одиницю як число n_{HV} , так і n_{VH} . Таким чином, верхня межа визначається

нерівністю $n_{HV} = n_{VH} \leq c$. Для визначення нижньої межі зауважимо, що шлях завдовжки m ФМ повинен включати по два або більше ФМ, щонайменше з $(m-c)/(r-1)$ стовпців \square .

Наприклад розглянемо мережу, зображену на рис. 3.9 б, для якої $c = r = 3$ та $m = 8$. Отже, $n_{HV} = n_{VH} \leq 3$. Використовуючи доведену теорему, можна визначити значення n_{HH} та n_{VV} очевидно, що $n_{HH} = n_{VV} - c$. Звідси

$$n_{HH} \leq \frac{r \cdot c - m}{r - 1}, \quad n_{VV} = m - n_{HH} - n_{HV} - n_{VH} = m - c - n_{HV}$$

Таким чином,

$$m - 2 \cdot c \leq n_{VV} \leq \frac{(m - c) \cdot (r - 2)}{r - 1}$$

для мережі, показаної на рис. 3.9, б, в, $n_{HH} = 0$ та $n_{VV} = 2$. Отримані граничні значення застосовуються під час перевірки коректності встановленої конфігурації ОС.

Основною перевагою двовимірної ОС з розподіленим управлінням реконфігурацією є те, що механізм реконфігурації перестав бути частиною ядра системи. Така структура має такі переваги:

- несправність механізму комутації рівноцінна несправності ФМ;
- робоча конфігурація підтримується виключно справними ФМ і, в такий спосіб, залежить від станів несправних ФМ.

Єдиним вузлом, який може впливати несправний ФМ, є вихідна шина. Цей вплив можна усунути за допомогою схеми вихідного селектора, як показано на рис. 3.11. Основними недоліками такої структури мережі є складність процедури реконфігурації і те, що мережа з s резервними модулями буде стійка до несправностей, що мають кратність, меншу s (тобто $f < s$).

Несправний модуль (ΦM_6) ізолюється разом з двома модулями-наступниками його сигналів (ΦM_5 і ΦM_7)

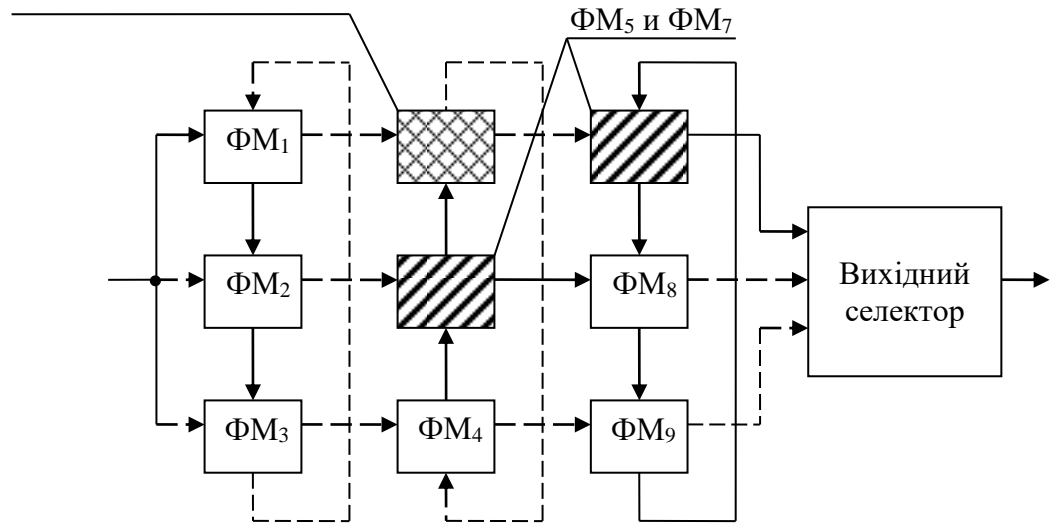


Рисунок 3.11 – Структура ІКМ з модулем вихідного селектора

Нижче наведено алгоритм процедури реконфігурації ІКМ. В алгоритмі використовуються чотири вектори даних: 1) вектор – $vx_шини$ і вектор – $вих_шини$ що задають вхідні та вихідні шини справних ФМ; 2) вектор – маршрут, що представляє поточну конфігурацію ІКМ. Цей вектор визначає номери функціонуючих модулів у маршруті обробки даних; 3) вектор - матриця наступників розмірністю $2 \times n$, де n – загальна кількість ФМ, в якій i -й стовбець задає два наступники кожного i -го ФМ.

Вхідним параметром, визначальним початок виконання алгоритму реконфігурації, номер несправного модуля. У процесі виконання алгоритму змінюються значення векторів: $vx_шини$, $вих_шини$, маршруту, матриці наступників, і навіть ініціюється процес пересилання даних.

Алгоритм реконфігурації мережі може бути представлений у вигляді послідовності наступних кроків.

Алгоритм 3.2.

Крок 1. Виключити несправний ФМ із векторів `vx_шини` та `вих_шини`.

Крок 2. Якщо хоча б один із векторів `vx_шини` та `вих_шини` не містить жодного справного ФМ, виконати функцію `відмова_системи` і перейти до кроку 9. В іншому випадку перейти до кроку 3 алгоритму.

Крок 3. Модифікувати матрицю наступників шляхом зміни всіх значень 1.

Крок 4. Виконати функцію `пошук_маршруту` для знаходження всіх маршрутів, що мають довжину, що відповідає числу справних ФМ. Результат подати як матриці `все_маршрути`, у кожному рядку якої описаний окремий маршрут.

Крок 5. Видалити з матриці `все_маршрути`, маршрути, останній ФМ яких включено до складу вектора `вих_шини`.

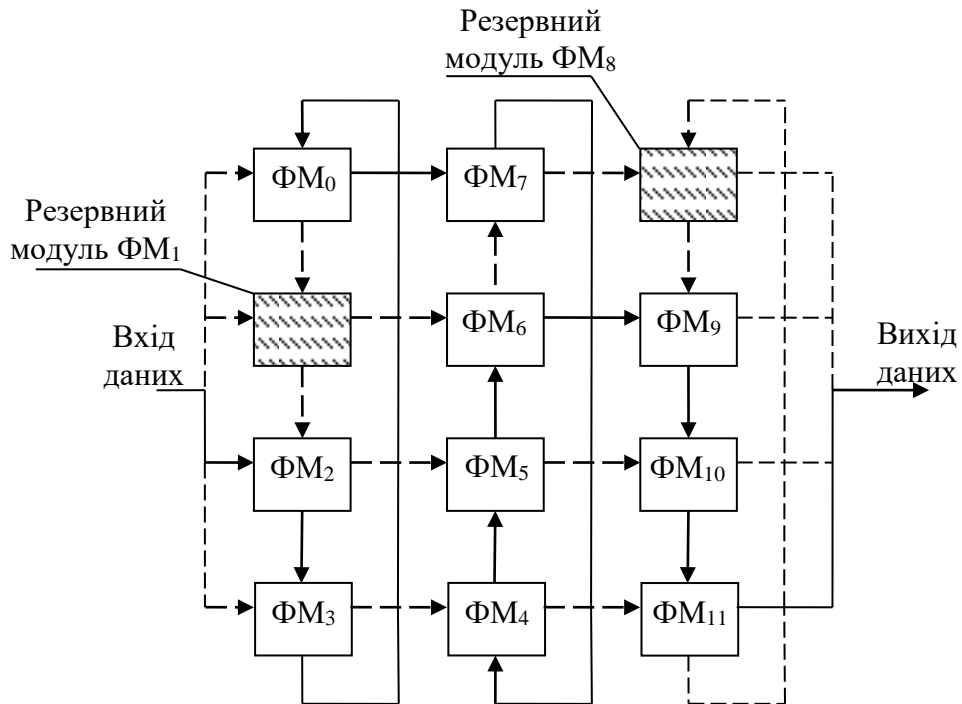
Крок 6. Якщо матриця `все_маршрути` не містить жодного ФМ, виконати функцію `відмова_системи` і перейти до кроку 9. Інакше перейти до кроку 7 алгоритму.

Крок 7. Вибрати в матриці `все_маршрути` - маршрут, якому відповідає найменша кількість пересилань даних.

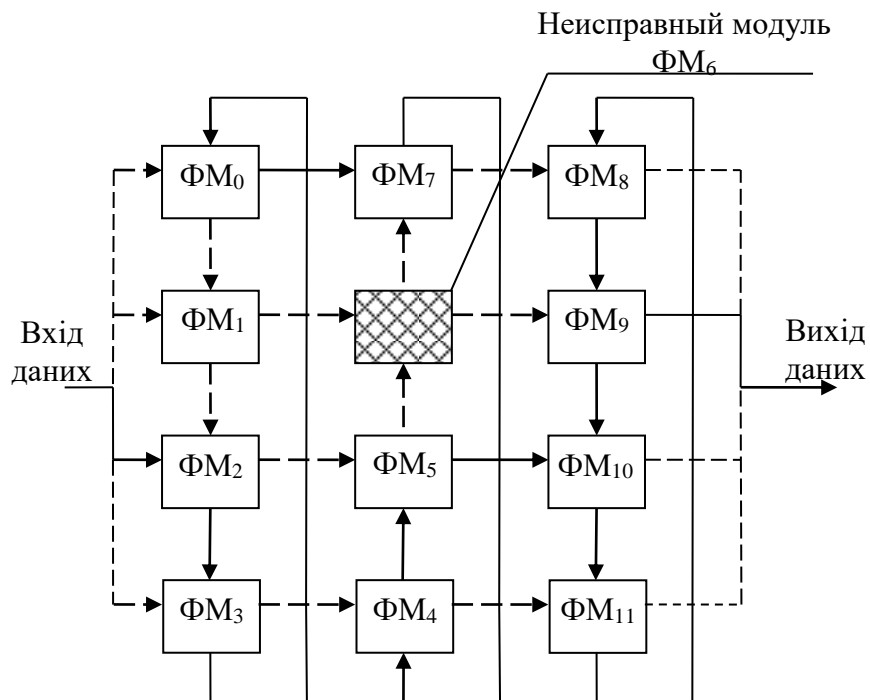
Крок 8. Виконати функцію пересилання даних між ФМ. У разі руйнування даних виконати процедуру відновлення шляхом повторного виконання фрагментів програми. Кінець алгоритму.

Приклад 3.1. Нехай задана двомірна мережа розмірністю, в якій резервні ФМ розташовані у клітинах {1, 8}. У вихідній конфігурації операційної системи використовується структура, подана на рис. 3.12, а.

Якщо СТД виявила несправні ФМ₆, то відповідно до алгоритму 3.2 мережа реконфігурується в структуру (рис. 3.12 б), в якій обраний маршрут обробки даних з виключенням ФМ₆ і підключенням ФМ₈. При цьому $n = 10$ модулів, що функціонують, залишилося таким же як у вихідній конфігурації, що зберігає її продуктивності.



а) вихідна конфігурація мережі за маршрутом (ФМ₂, ФМ₃, ФМ₀, ФМ₇, ФМ₄, ФМ₅, ФМ₆, ФМ₉, ФМ₁₀, ФМ₁₁)



б) структура реконфігурованої мережі з несправним ФМ₆ з маршрутом (ФМ₂, ФМ₃, ФМ₀, ФМ₇, ФМ₄, ФМ₅, ФМ₁₀, ФМ₁₁, ФМ₈, ФМ₉)

Рисунок 3.12 – Структура двомірної мережі для прикладу 3.1

3.3 Діагностування та верифікація програмних модулів HDL-моделей

В якості прикладу використання ІКМ розглянемо обчислення тестопридатності для HDL-моделі (Verilog) дискретного Вейвлет-перетворення, математична модель якого представлена на рис.3.13

$$\psi_{m,n} = a_0^{-m/2} \psi \left(\frac{t - nb_0}{a_0^m} \right)$$

Рисунок 3.13 – Математична модель дискретного Вейвлет-перетворення

Вейвлет-перетворення широко використовується для аналізу сигналів. Крім цього, він знаходить велике застосування в області стиснення даних. У дискретному вейвлет-перетворенні найбільша інформація в сигналі міститься при високих амплітудах, а менш корисна - при низьких. Стиснення даних може бути отримано за рахунок відкидання низьких амплітуд. Вейвлет-перетворення дозволяє отримати високе співвідношення стиснення у поєднанні з гарною якістю відновленого сигналу.

Тестопридатність обчислюється за методикою, запропонованою проф. Хахановим [15] на основі архітектурної моделі Вейвлет-перетворення. Далі представлені компоненти інфраструктури для верифікації та діагностичного обслуговування двох модулів WT SoC, а саме: транзакційні графи та побудовані на їх основі логічні функції тестопридатності, керованості та спостережуваності HDL-моделей; графіки оцінювання тестопридатності всіх вершин та всіх графів програмного HDL-коду; виконані діагностичні експерименти, які підтверджують ефективність та валідність запропонованих моделей та методу пошуку дефектів

Структурний аналіз графа для визначення тестопридатності дає можливість створити транзакційний граф, представлений на рисунку 3.14, який має вхідні шини; вихідні шини; логічні змінні; реєстрові змінні; вектори; пам'ять.

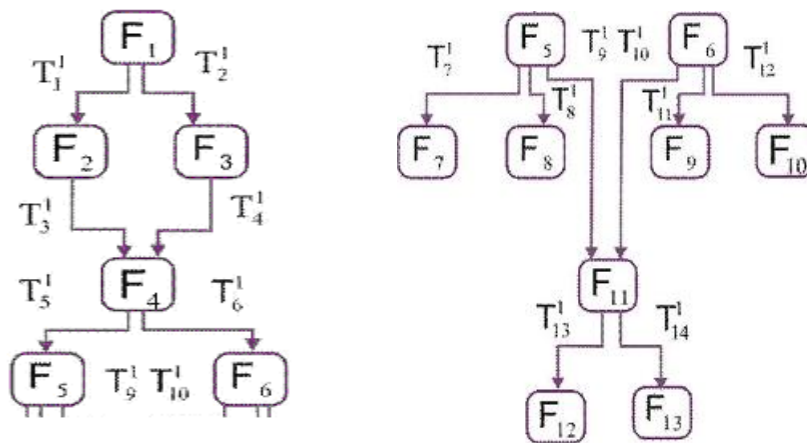


Рисунок 3.14 – Транзакційний граф HDL-кода моделі Вейвлет-перетворення

Логічні функції керованості та спостережуваності кожної вершини транзакційного графа записуються у вигляді кон'юнкції диз'юнктивних термів:

$$U_i = \frac{1}{T} \left(\bigwedge_{j=1}^{X_1} T_{1j}^X \right) \dots \left(\bigwedge_{j=1}^{X_{i-1}} T_{i-1j}^X \right) \left(\bigvee_{i=1}^{X_i} T_{ij}^X \right) \quad N_i = \frac{1}{T} \bigwedge_{j=1}^{X_i} T_j^i \left(\bigvee_{i=1}^{X_i} T_j^i d_i^X \vee t_i^X \right).$$

На основі транзакційного графа будуються логічні функції керованості та спостережуваності, приклад логічної функції керованості входів та спостереженості виходів для вершини представлені нижче:

а) керованість входів

$$F_2 = T_1^1; \quad F_3 = T_2^1; \quad F_4 = T_1 T_3 \vee T_4 T_2; \quad F_5 = T_1^1 T_5^1 T_3^1 \vee T_5^1 T_4^1 T_2^1; \quad F_6 = T_6^1 T_1^1 T_3^1 \vee T_6^1 T_4^1 T_5^1;$$

$$F_7 = T_7^1 T_1^1 T_5^1 T_3^1 \vee T_7^1 T_5^1 T_4^1 T_2^1; \quad F_8 = T_8^1 T_1^1 T_5^1 T_3^1 \vee T_8^1 T_5^1 T_4^1 T_2^1; \quad F_9 = T_{11}^1 T_6^1 T_1^1 T_3^1 \vee T_{11}^1 T_6^1 T_4^1 T_5^1;$$

$$F_{10} = T_{12}^1 T_6^1 T_1^1 T_3^1 \vee T_{12}^1 T_6^1 T_4^1 T_5^1; \quad F_{11} = T_{10}^1 T_6^1 T_1^1 T_3^1 \vee T_{10}^1 T_6^1 T_4^1 T_5^1 \vee T_9^1 T_5^1 T_1^1 T_3^1 \vee T_9^1 T_5^1 T_4^1 T_2^1;$$

$$F_{12} = T_{13}^1 T_{10}^1 T_6^1 T_1^1 T_3^1 \vee T_{13}^1 T_{10}^1 T_6^1 T_4^1 T_5^1 T_{13}^1 \vee T_{13}^1 T_9^1 T_5^1 T_1^1 T_3^1 \vee T_{13}^1 T_9^1 T_5^1 T_4^1 T_2^1;$$

$$F_{13} = T_{14}^1 T_{10}^1 T_6^1 T_1^1 T_3^1 \vee T_{14}^1 T_{10}^1 T_6^1 T_4^1 T_5^1 \vee T_{14}^1 T_9^1 T_5^1 T_1^1 T_3^1 \vee T_{14}^1 T_9^1 T_5^1 T_4^1 T_2^1.$$

б) спостережуваність виходів

$$F_{11} = T_{13}^1 \vee T_{14}^1; \quad F_5 = T_7^1 \vee T_8^1 \vee T_9^1 T_{13}^1 \vee T_9^1 T_{14}^1; \quad F_6 = T_{11}^1 \vee T_{12}^1 \vee T_{10}^1 T_{13}^1 \vee T_{10}^1 T_{14}^1;$$

$$F_4 = T_5^1 T_7^1 \vee T_5^1 T_8^1 \vee T_5^1 T_9^1 T_{13}^1 \vee T_5^1 T_9^1 T_{14}^1 \vee T_6^1 T_{11}^1 \vee T_6^1 T_{12}^1 \vee T_6^1 T_{10}^1 T_{13}^1 \vee T_6^1 T_{10}^1 T_{14}^1;$$

$$\begin{aligned}
 F_3 &= T_4^1 T_5^1 T_7^1 \vee T_4^1 T_5^1 T_8^1 \vee T_4^1 T_5^1 T_9^1 T_{13}^1 \vee T_4^1 T_5^1 T_9^1 T_{14}^1 \vee T_4^1 T_6^1 T_{11}^1 \vee T_4^1 T_6^1 T_{12}^1 \vee T_4^1 T_6^1 T_{10}^1 T_{13}^1 \vee T_4^1 T_6^1 T_{10}^1 T_{14}^1; \\
 F_2 &= T_3^1 T_5^1 T_7^1 \vee T_3^1 T_5^1 T_8^1 \vee T_3^1 T_5^1 T_9^1 T_{13}^1 \vee T_3^1 T_5^1 T_9^1 T_{14}^1 \vee T_3^1 T_6^1 T_{11}^1 \vee T_3^1 T_6^1 T_{12}^1 \vee T_3^1 T_6^1 T_{10}^1 T_{13}^1 \vee T_3^1 T_6^1 T_{10}^1 T_{14}^1; \\
 F_1 &= T_2^1 T_4^1 T_5^1 T_7^1 \vee T_2^1 T_4^1 T_5^1 T_8^1 \vee T_2^1 T_4^1 T_5^1 T_9^1 T_{13}^1 \vee T_2^1 T_4^1 T_5^1 T_9^1 T_{14}^1 \vee T_2^1 T_4^1 T_6^1 T_{11}^1 \vee T_2^1 T_4^1 T_6^1 T_{12}^1 \vee T_2^1 T_4^1 T_6^1 T_{10}^1 T_{13}^1 \vee \\
 &\vee T_2^1 T_4^1 T_6^1 T_{10}^1 T_{14}^1 \vee T_1^1 T_3^1 T_5^1 T_7^1 \vee T_1^1 T_3^1 T_5^1 T_8^1 \vee T_1^1 T_3^1 T_5^1 T_9^1 T_{13}^1 \vee T_1^1 T_3^1 T_5^1 T_9^1 T_{14}^1 \vee T_1^1 T_3^1 T_6^1 T_{11}^1 \vee T_1^1 T_3^1 T_6^1 T_{12}^1 \vee \\
 &\vee T_1^1 T_3^1 T_6^1 T_{10}^1 T_{13}^1 \vee T_1^1 T_3^1 T_6^1 T_{10}^1 T_{14}^1.
 \end{aligned}$$

Якість програмного продукту на основі аналізу графа транзакцій залежить від керованості, спостережуваності, а також додаткових витрат на формування функціональностей, тесту, асерцій. Керованість (спостережуваність є функція від числа операторів, що входять до вершини або виходять з вершини) транзакційного графа, а також від структурної глибини вершини, що розглядається, щодо вхідної (вихідної) шини.

Для реалізації обчислень будується ІКМ, яка відповідає транзакційному графу і реалізує необхідні обчислення (рис 3 15).

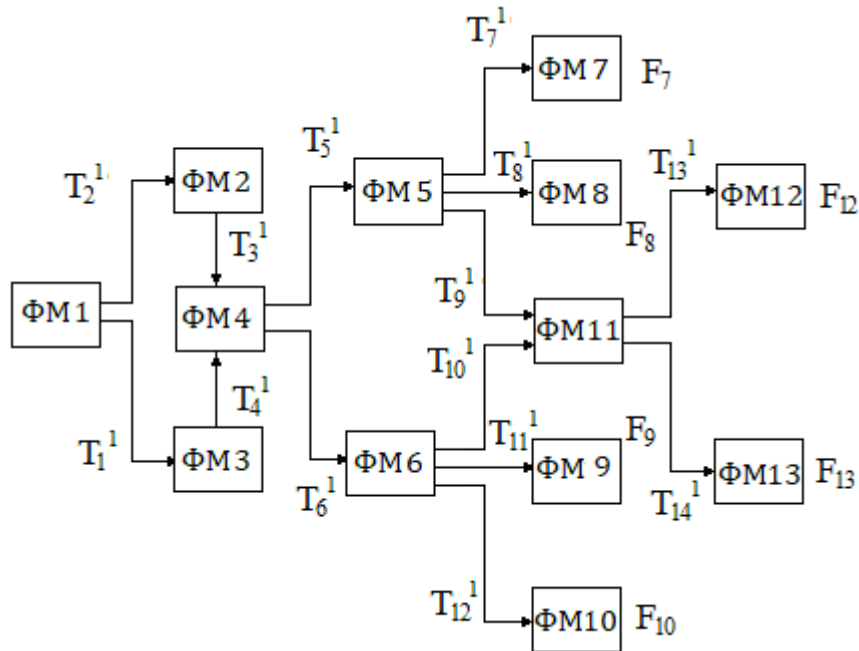


Рисунок 3.15 – ІКМ обчислення керованості та спостережуваності HDL-кода моделі Вейвлет-перетворення

Обчислення логічних функцій керованості та спостережуваності обробляються на основі використання формули для визначення якості програмного продукту на основі аналізу графа транзакцій, і отримуємо мінімальні оцінки вершин у частині спостережуваності, для яких будуються асерції як додаткові точки моніторингу програмного коду [15].

Відповідно до значень параметрів тестопридатності на рисунку 3.16 представлені графіки керованості, спостережуваності всіх вершин транзакційного графа головної програми, яка візуально ілюструє критичні точки спостереження, цікаві для встановлення в них асерцій.

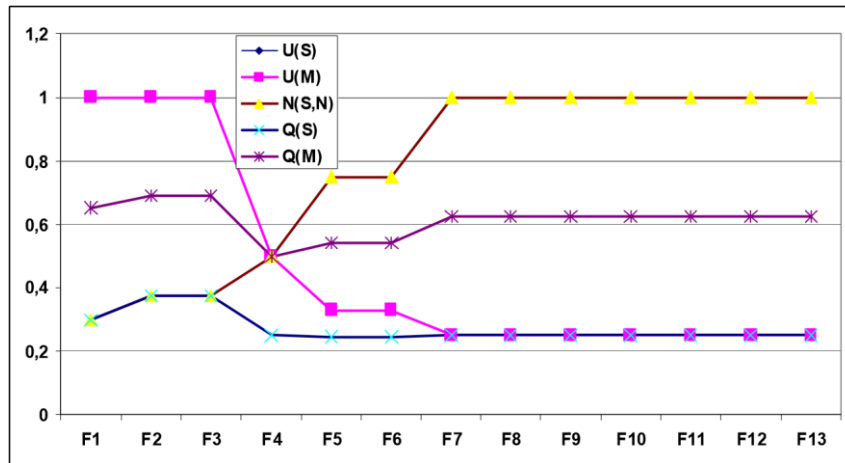


Рисунок 3.16 – Графіки параметрів тестопридатності Xilinx моделі

Загальна оцінка тестопридатностей двох видів графа (мульти- та одиночні дуги) має вигляд: $Q(F_0) = \{Q(F_0^S) = 0,33766; Q(F_0^M) = 0,61525\}$. Отримані оцінки є суттєвими у порівнянні кількох варіантів проекту, зроблених різними розробниками, зрозуміло, що проект, який має велику інтегральну оцінку тестопридатності, приймається до реалізації.

У першому випадку критерій якості дорівнює одиниці, що свідчить про наявність даного дефекту в програмному коді. У другому випадку, коли критерій не дорівнює одиниці, можлива присутність будь-якого поєднання з трьох несправностей. Надамо діагностичне забезпечення одного із програмних модулів нижнього рівня, який є компонентом транзакційного

графа програми. На рисунку 3.17 наведено логічні функції керованості і спостережуваності на основі транзакційного графа.

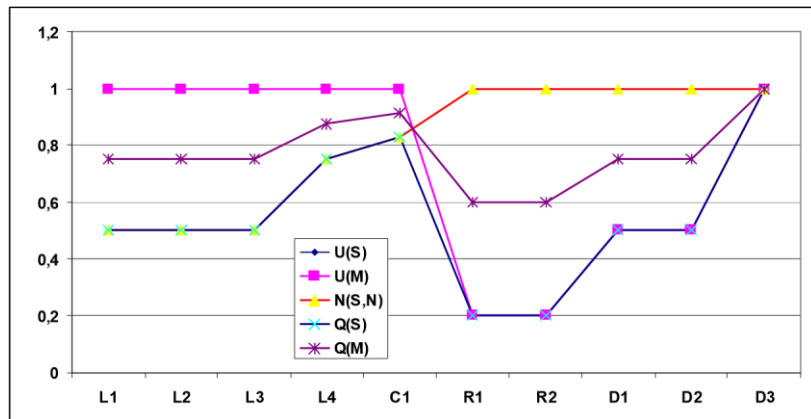


Рисунок 3.17– Графіки параметрів тестопридатності Row_buffer Xilinx моделі

Для критичних точок, визначених у результаті аналізу тестопридатності транзакційного графа, розроблена асерційна модель перевірки основних характеристик дискретного косинусного перетворення. Перевірка моделі або перевірка властивості полягає у вичерпній і автоматичній перевірці чи задана модель певної системи відповідає заданій специфікації. Зазвичай, йдеться про апаратну або програмну систему, а специфікація містить вимоги безпеки як-от відсутність взаємних блокувань та інших подібних критичних станів, що можуть призвести до аварії. Перевірка моделі – це техніка для автоматичної перевірки правильності властивостей системи зі кінцевою кількістю станів. Для того, що розв'язати таку задачу алгоритмічно, модель системи й специфікація формулюються якоюсь точною математичною мовою. Для цього проблему записують як задачу в логіці, – перевірити чи задана структура задовольняє заданій логічній формулі. Ця загальна концепція застосовна для багатьох видів логік і схожих структур. Простим прикладом може бути перевірка чи задана формула в численні висловлень, задоволених заданою структурою.

ВИСНОВКИ

У процесі виконання магістерської кваліфікаційної роботи, була досягнута мета роботи, а саме: розроблені методи підвищення тестопридатності моделей реконфігурованих інтерактивних обчислювальних мереж на програмованих функціональних модулях а також методики проведення діагностичних експериментів над моделями зазначених мереж.

Для досягнення мети роботи, були виконані теоретичні етапи. З метою оцінки складності процесу діагностування дискретних пристроїв (ДП) з елементами пам'яті проведений аналіз методів побудови діагностичних експериментів, заснованих на функціональному підході і використанні автоматних моделей ДП.

Проведені у поданій магістерській кваліфікаційній роботі дослідження, дозволяють зробити висновки щодо отриманих наукових результатів інноваційних технологій підвищення тестопридатності реконфігурованих інтерактивних комп'ютерних мереж на програмованих функціональних модулях.

Запропоновано процедуру модифікації структури функціональних комірки мережі шляхом введення додаткового входу, що спрощує процедуру синтезу перевіряючих тестів для ОІКМ із бічними виходами, що спостерігаються, з метою покращення показників керованості ОІКМ у процесі діагностичного експерименту.

Розроблено процедуру побудови діагностичного експерименту для ІКМ із виходами, що спостерігаються, у якій автоматна модель ФМ мережі є сильнозв'язним автоматом і має відрізняючу послідовність. Показано, що така мережа є тестопридатною.

Розроблено метод та процедуру синтезу одновимірних ІКМ з резервними ФМ, засновані на централізованому управлінні процедурою реконфігурації ІКМ при виявленні несправного ФМ та використанні

шунтуючих комутаційних мереж. Вирішено завдання оптимального розміщення резервних ФМ у структурі ІКМ, на основі критерію мінімальності числа пересилань даних між ФМ, що забезпечує мінімальність часових витрат на відновлення працездатності мережі без деградації продуктивності.

Розроблено метод та процедуру синтезу одновимірних та двовимірних ІКМ з розподіленим управлінням реконфігурацією із вбудованою у ФМ системою внутрішньої комутації вхід - вихідних шин ФМ. Визначено та обґрунтовано нижні та верхні межі довжина маршрутів у двовимірній ІКМ, які використовувалися для вибору альтернативного варіанта реконфігурації при розробці алгоритмічної процедури заміни несправного ФМ резервним та відновлення працездатності ІКМ, що виключає необхідність вирішення оптимізаційної задачі вибору найкращого варіанта реконфігурації обробки даних у ІКМ.

Побудовано логічні функції керованості та спостережуваності для реального прикладу HDL-моделі і графіків; відповідні їм, критичні точки, які дають можливість проаналізувати шляхи подальшого поліпшення коду проекту шляхом установки асерцій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мірошник М.А. Проектування діагностичної інфраструктури обчислювальних систем і пристроїв на ПЛІС: монографія / М.А. Мірошник. – Харків.: ХУПС, 2012. – 188 с.
2. Aladjev V.Z. Classical Homogeneous Structures. Cellular Automata / V.Z. Aladjev. – Fultus Publishing, 2009. – 535 p.
3. Мірошник М.А. Синтез часових автоматів з операційним перетворенням коду станів. / Мірошник М.А., Зайченко О.Б., Мірошник А. М., Зайченко Н.С. // Modern scientific researches, №12, 2021, С.65-79.
4. Miroshnyk, M.A. Design timed FSM with VHDL Moore pattern / Miroshnyk, M.A.; Shkil, O.S; Kulak, E.N.;, Rakhlis, D.Y.; Miroshnyk, A.M.; Malahov, N.V. //Journal Radio electronics computer science control, ISSN 1607-3274 eISSN 2313-688X, 2020, Issue 2, P. 137-148.
5. Bruck J. Wildcard Dimensions, Coding Theory and Fault-Tolerant Meshes and Hypercubes/ J. Bruck, R. Cypher, C. Ho // IEEE Trans. Computers – 1995. – Vol. 44. – P. 150 – 155.
6. Miroshnik M.A. Methods for designing self-checking digital machines. / M. A. Miroshnik, E. N. Kulak, E. M. Aliyeva, D. G. Karaman, Yu. V. Pakhomov // Telecommunications and Radio Engineering. – 2017. – Volume 76, Issue 15, 2017, P. 1367-1377.
7. Николайчук, Я.М. Структуризація, методи та моделі інтерактивної взаємодії оператор-інформаційна система моніторингу об'єктів нафтогазової галузі / Я.М. Николайчук, Н.Я. Возна, Г.Я. Процюк та ін. // Розвідка та розробка нафтових і газових родовищ: всеукр. наук.-техн. журнал. – Івано-Франківськ, 2015. – № 2 (55). – С. 111-118.
8. Николайчук, Я.М. Дослідження системних функцій та архітектури інтерактивних комп'ютерних мереж [Ел. ресурс] / Я.М. Николайчук, Р.В. Цанько, Н.Я. Возна // Вісник Хмельницького національного університету. – Хмельницький, 2012. – № 4 (191). – С. 73-78.

9. Кузьо М.М. Реконфігуровані обчислювальні системи на однорідній структурі Lviv Polytechnic National University Institutional Repository <http://ena.lp.edu.ua>, CSN., Number 688 2010; С.152-157.

10. Мирошник М.А. Однородные сети с распределенной системой реконфигураций / М.А. Мирошник, Я.Ю. Королева, Л.В. Дербунович // Вісник НТУ „ХПІ”. Збірник наукових праць. Тематичний випуск: Автоматика та приладобудування. – Харків: НТУ „ХПІ”. – 2010. – №20. – С. 71-78.

11. Fridman A.D., Menon P.R. Fault detection in digital circuits. – New Jersey: Prentice Hall, 1971. – 220 p.

12. Самофалов К.Г. Основы теории многоуровневых конвейерных вычислительных систем / К.Г. Самофалов, Г.М. Луцкий. – М.: Радио и связь, 1989. – 272 с.

13. Wang C. Fault Detection with multiple observers / C. Wang, M. Schwart // IEEE. ACM Trans. Networking. – 1993. – №1. – P. 48 – 55.

14. Shkil A. Assertion based design of timed finite state machine / A. Shkil, A. Miroschnyk, G. Kulak, K. Pshenychnyi // Proceedings of 2021 IEEE East-West Design & Test Symposium (EWDTS'21), September 10-13, Batumi, Georgia, 2021. – P. 291-294.

15. Хаханов В.И. Метод верификации HDL-кода на основе транзакционного логического графа / В.И. Хаханов, И.А. Побеженко, В.А. Василенко, С.В. Чумаченко // АСУ и приборы автоматики : всеукр. межвед. науч.-техн. сб. – Х. : Изд-во ХНУРЭ, 2009. – Вып. 148. – С. 87–101.

16. Шкіль О.С. Методи побудови тестів для інтерактивних комп'ютерних мереж на структурно-логічному рівні / М.А. Мірошник, О.С. Шкіль, Д.Ю. Рахліс, А.М. Мірошник, Д.А. Лобойченко // Вісник Національного технічного університету "ХПІ". – 2023. – № 1–2 (9–10). – С. 74-83.

