

УДК 519.711; 004.421; 519.68  
КП  
№ держреєстрації 0111U002956  
Інв. №

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ  
(ХНУРЕ)

61166, м. Харків, пр. Леніна, 14  
тел. (057) 70-21-425

ЗАТВЕРДЖУЮ  
Проректор з наукової роботи ХНУРЕ  
д-р фіз.-мат. наук, проф.

\_\_\_\_\_ М.І.Сліпченко  
«\_\_» \_\_\_\_\_ 2014 р.

ЗВІТ  
ПРО НАУКОВО-ДОСЛІДНУ РОБОТУ № 259

**МУЛЬТИПРОЦЕСОРНА СИСТЕМА ПОШУКУ, РОЗПІЗНАВАННЯ ТА  
ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ІНФОРМАЦІЙНОЇ КОМП'ЮТЕРНОЇ  
ЕКОСИСТЕМИ**

(заключний)

Керівник НДР  
декан факультету КІУ ХНУРЕ,  
д.т.н., проф.

В.І. Хаханов

2014

Рукопис закінчено «\_\_» \_\_\_\_\_ 2014 р.

Результати роботи розглянуті науково-методичною радою ХНУРЕ,  
Протокол № \_\_ від «\_\_» \_\_\_\_\_ 2014 р.

## СПИСОК АВТОРІВ

Керівник НДР, декан факультету КІУ,  
д.т.н., проф.

Відповідальний виконавець НДР,  
завідувач кафедри АПОТ, д.т.н.,  
проф.,

Проф. каф. АПОТ, д.т.н.

Проф. каф. АПОТ, д.т.н.

С.н.с. каф. АПОТ, к.т.н.

Н.с. каф. АПОТ, к.т.н.

Доц. каф. АПОТ, к.т.н.

Доц. каф. АПОТ, к.т.н.

Мл. наук, співроб. каф. АПОТ

Доц. каф. АПОТ, к.т.н.

Аспірант каф. АПОТ

Аспірант каф. АПОТ

Проф. каф. АПОТ, к.т.н.

Проф. каф. АПОТ, д.т.н.

Пошукач каф. АПОТ

Аспірант каф. АПОТ

Доц. каф. АПОТ, к.т.н.

Проф. каф. АПОТ, к.т.н.

Аспірант каф. АПОТ

Аспірант каф. АПОТ

Аспірант каф. АПОТ

В.І. Хаханов

(розділи 1-6)

С.В. Чумаченко

(розділи 1,2,6)

Є.І. Литвинова (розділ 3-5)

Г.Ф. Кривуля (розділ 3,6)

В.Б. Таранов (розділ 3)

Д.Є. Шуклін (розділ 3)

О.С. Шкиль (розділ 3,6)

І.В. Філіпенко (розділ 4,6)

А.А. Давидов (розділ 3,6)

Г.В. Хаханова (розділ 4,6)

О.С. Міщенко (розділ 1,6)

В.В. Варєца (розділи 2,6)

М.Я. Какурін (розділ 4)

І.В. Хаханова (розділ 4)

К.Є. Герасименко (розділ 5)

О.О. Горобець (розділ 3,6)

Г.П. Фастовець (розділ 3,6)

В.П. Немченко (розділ 3,6)

Д.Ю. Кучеренко (розділ 3,6)

А.Н. Зіарманд (розділ 3,6)

С.О. Дементьев (розділ 2, 6)

Нормоконтролер

Т.Г. Силантьєва

## РЕФЕРАТ

Звіт про НДР: 192 с., 46 рис., 7 табл., 121 джерел.

публікації: монографії 2, навчальні посібники та підручники 1, статті 36, тези 72, патенти 1.

Мета НДР – істотне ( $\times 100$ ) підвищення швидкодії процедур пошуку, розпізнавання й прийняття рішень шляхом мультипроцесорної й паралельної реалізації асоціативно-логічних векторних операцій для аналізу графових і табличних структур даних у дискретному булевому просторі без використання арифметичних операцій.

Об'єкт дослідження – інфраструктура пошуку, розпізнавання та прийняття рішень у дискретному булевому просторі на основі використання алгебри векторної логіки, мультипроцесорної платформи аналізу асоціативно-логічних структур даних і неарифметичного інтегрального критерію якості.

Методи дослідження – булева алгебра, векторно-асоціативна логіка, теорія множин, теорія графів, теорія цифрових автоматів – для побудови моделей тестування; логічний аналіз, теорія алгоритмів, методи проектування та моделювання цифрових систем – для синтезу тестів, структур даних і сервісного обслуговування; методи аналізу якості моделей та ефективності сервісного обслуговування цифрових систем – для досягнення заданої глибини пошуку дефектів запропонованими методами; промислові засоби логічного синтезу схем та аналізу функціонального покриття, а також генерації тестів – для створення програмно-апаратної інфраструктури діагностування SoC.

За завданням НДР проведено аналіз стану проблеми, висвітлені основні задачі та проблемні питання, означені шляхи їх реалізації, наведено характеристику основних наукових та практичних результатів роботи. Наукові-практичним результатом, на одержання якого була спрямована НДР, є розробка та впровадження методів та моделей, орієнтованих на створення мультипроцесорної системи пошуку, розпізнавання та прийняття рішень для інформаційної комп'ютерної екосистеми, які базуються не на копіюванні живого мозку людини у неживому кремнієвому кристалі, але на інкапсулюванні функціональності мозку в кристал, що надає можливість поширити міцність сервісів, корисних для розв'язування практичних задач в інформаційному просторі.

Ключові слова: мультипроцесор, інформаційна комп'ютерна екосистема, інформаційний простір, асоціативно-логічні структури даних, процес-моделі пошуку, розпізнавання й вибору рішення.

## ЗМІСТ

Перелік умовних позначень	6
Передмова	7
1 Алгебра векторно-матричної логіки	9
1.1 Система законів алгебри векторно-матричної логіки та їх доведення	10
1.2 Висновки	25
2 Векторний логічний аналіз діагностичної інформації	27
2.1 Інфраструктура формування умов та прийняття рішень	27
2.2 Процес-моделі аналізу асоціативних структур даних	31
2.3 Метрика алгебри векторної логіки	44
2.4 Аналіз табличних структур даних	54
2.5 Спрощення критерію якості	56
2.6 Висновки	60
3 Моделі генерації тестів і методи діагностування SOC-компонентів	61
3.1 Моделі генерації тестів для функціональних елементів	62
3.2 Синтез тестів $F \oplus L$ -методом	65
3.3 Апарат булевих похідних для синтезу тестів	68
3.4 Векторно-логічні методи аналізу таблиць несправностей	77
3.5 3D-Метод діагностування дефектних компонентів у SoC	82
3.6 Мультирівнева модель і метод (движок) діагностування цифрової системи	92
3.7 Висновки	95
4 Інфраструктура логічного асоціативного мультипроцесора	97
4.1 Вступ	97
4.2 Скалярна метрика для оцінювання розв'язку	99
4.3 Архітектура логічного асоціативного мультипроцесора	104
4.4 Інфраструктура векторно-логічного аналізу	112
4.5 Апаратна імплементація компонентів мультипроцесора	120



4.6 Висновки	128
5 Метод підвищення тестопридатності процесорних систем з використанням функціональних елементів на базі арифметичних операцій	130
5.1 Математичний апарат методу підвищення тестопридатності процесорних систем	130
5.2 Модифікація дедуктивного методу пошуку несправностей	133
5.3 Аналіз тестопридатності	139
5.4 Висновки	144
6 Інші результати, отримані та реалізовані в рамках НДР	146
6.1 Результативність виконання НДР	146
6.2 Монографії та навчальні посібники	147
6.3 Патенти	147
6.4 Наукові статті	148
6.5 Тези доповідей на конференціях	152
6.6 Договори з зацікавленими організаціями	163
6.7 Організація і проведення конференцій та семінарів	163
6.8 Виконані інноваційні проекти та проекти, що виконуються	166
6.9 Отримані гранти	167
6.10 Теми захищених дисертацій	169
6.11 Виставки	170
6.12 Теми захищених магістерських робіт за тематикою НДР	171
6.13 Залучення позабюджетних коштів	174
6.14 Розробки	175
6.15 Використання результатів НДР у навчальному процесі	179
Список посилань	180

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

LAMP	– Logical Associative MultiProcessor
ЛАМП	– Логічний асоціативний мультипроцесор
slc	– Shift Left Bit Crowding
LUT	– Look Up Table
PLD	– Programmable Logic Device
CPU	– Central Processor Unit
НВІС	– Надвелика інтегральна схема
ППУ	– Пристрій послідовного управління
LP	– логічний процесор
UUT	– Unit Under Test
MUT	– Model Under Test
AD	– Address Decoder
SM	– Spare Memory
VLAS	– Vector Logical Algebra Space
АЦП	– Аналого-цифровий перетворювач
ЦАП	– Цифро-аналоговий перетворювач
ПЦОВ	– Пристрій цифрової обробки відеосигналу
СВІ	– Система відображення інформації
СФР	– Система функціональної регенерації
АП	– Адресний процесор
ЗП	– Запам'ятовуючий пристрій
БР	– Буфер регенерації
ФГА	– Функціональний генератор адрес
ПК	– Персональний комп'ютер

## ПЕРЕДМОВА

*Мета НДР* – істотне (x100) підвищення швидкодії процедур пошуку, розпізнавання й прийняття рішень шляхом мультипроцесорної й паралельної реалізації асоціативно-логічних векторних операцій для аналізу графових і табличних структур даних у дискретному булевому просторі без використання арифметичних операцій.

*Об'єктом дослідження* є інфраструктура пошуку, розпізнавання та прийняття рішень у дискретному булевому просторі на основі використання алгебри векторної логіки, мультипроцесорної платформи аналізу асоціативно-логічних структур даних і неарифметичного інтегрального критерію якості.

*Предмет дослідження* – асоціативно-логічні структури даних і процес-моделі пошуку, розпізнавання й вибору розв'язок на основі неарифметичного інтегрального критерію якості шляхом використання мультипроцесорної системи на кристалі, що оперує векторними логічними операціями.

Основна ідея проекту полягає в розробці та впровадженні методів та моделей, орієнтованих на створення мультипроцесорної системи пошуку, розпізнавання та прийняття рішень для інформаційної комп'ютерної екосистеми, які базуються не на копіюванні живого мозку людини у неживому кремнієвому кристалі, але на інкапсулюванні функціональності мозку в кристал, що надає можливість поширити міцність сервісів, корисних для розв'язування практичних задач в інформаційному просторі. Підвищення швидкодії процедур пошуку, розпізнавання й прийняття рішень досягається шляхом мультипроцесорної й паралельної реалізації асоціативно-логічних векторних операцій для аналізу графових і табличних структур даних у дискретному булевому просторі без використання арифметичних операцій.

*Завдання проекту:*

- 1) системи законів векторно-матричної алгебри логіки;
- 2) технології обчислювання метрики у булевому векторно-логічному

алгебраїчному просторі; 3) метод покриття для визначення мінімальних форм булевих функцій на основі процес-моделі аналізу матриці;

4) процес-моделі аналізу асоціативних таблиць;

5) математичний апарат – алгебра векторної логіки для паралельного використання векторних операцій на спеціалізованому мультипроцесорному пристрої;

6) опис примітивних процес-моделей прийняття рішень;

7) діаграма Хасе для мінімізації логічних функцій;

8) інфраструктура мультипроцесорної системи пошуку, розпізнавання та прийняття рішень для інформаційної комп'ютерної екосистеми.

Пропонується теорія, методи й структури даних для швидкодійного мультипроцесора паралельного аналізу інформації, представленої у вигляді аналітичних, графових і табличних форм асоціативних відносин, для пошуку, розпізнавання й прийняття рішень в  $n$ -мірному векторному дискретному просторі. Розглядаються векторно-логічні процес-моделі актуальних прикладних завдань, у тому числі – діагностування й відновлення працездатності компонентів цифрових систем на кристалах, якість розв'язку яких оцінюється неарифметичною метрикою взаємодії булевих векторів. Розробляються процес-моделі асоціативно-логічного аналізу інформації на основі швидкодійного мультипроцесора в  $n$ -мірному векторному дискретному просторі. Розробляється алгебраїчна структура, що регулює векторно-матричні перетворення в дискретному булевому просторі для аналізу інформації на основі логічних операцій над асоціативними даними.

# 1 АЛГЕБРА ВЕКТОРНО-МАТРИЧНОЇ ЛОГІКИ

Представлено алгебраїчну систему, орієнтовану на математичну підтримку моделей і методів векторного алгебрологічного аналізу кібернетичного простору при пошуку, розпізнаванні й прийнятті рішень [57, 58, 60, 68].

Інфраструктура векторних алгебрологічних перетворень у просторі VLAS (Vector Logical Algebra Space) представлена на рис. 1.1. Вхідна інформація може бути задана вектором  $m$ , матрицею  $A$  і булевими змінними  $C$ . В останньому випадку попередньо виконується векторизація змінних, а потім обробка отриманої асоціації. Вихідна інформація представлена у вигляді вектора  $m$ , матриці  $A$ , а також булевої змінної  $Y$ , отриманої в результаті девекторизації асоціативної впорядкованої послідовності  $C$ . Для аналізу матричних або векторних даних, при заданих вхідних умовах, що представляють матрицю або вектор, далі запропоновано алгебраїчну векторно-матричну структуру.

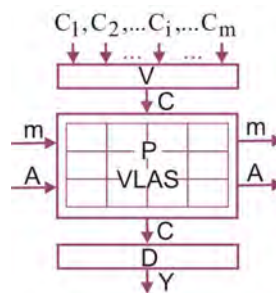


Рисунок 1.1 – Інфраструктура алгебраїчних перетворень

Алгебраїчні структури, регулюють векторно-матричні перетворення в дискретному векторному булевому просторі для аналізу інформації на основі логічних операцій над асоціативними даними. Представлено алгебру, орієнтовану на обробку пар (вектор - матриця).

## 1.1 Система законів алгебри векторно-матричної логіки та їх доведення

Елементами є  $n$ -мірні двійкові вектори виду:

$$m = (m_1, m_2, \dots, m_i, \dots, m_n), m_i \in \{0,1\}, i = \overline{1, n}. \quad (1.1)$$

і множини  $n$ -мірних двійкових векторів  $A$  виду:

$$A = \{a_s \mid a_s = (a_{s1}, a_{s2}, \dots, a_{si}, \dots, a_{sn}), a_{si} \in \{0,1\}, i = \overline{1, n}, s = \overline{1, p}\} \quad (1.2)$$

або в матричній формі:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1i} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2i} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{s1} & a_{s2} & \dots & a_{si} & \dots & a_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{p1} & a_{p2} & \dots & a_{pi} & \dots & a_{pn} \end{bmatrix}. \quad (1.3)$$

Потужність множини  $A$  задається за умовою кількістю векторів-рядків:  $|A| = p$ . Таким чином, можна розглядати як (1.2) матрицю (1.3) розміру  $p \times n$ .

Диз'юнкція й кон'юнкція вектора  $m$  та сукупності векторів  $A$  визначаються як послідовне виконання покомпонентного логічного додавання (множення) вектора  $m$  з кожним вектором-рядком з матриці  $C$ :

$$m \vee C = \{m \vee c_s \mid m \vee c_s = (m_1 \vee c_{s1}, m_2 \vee c_{s2}, \dots, m_i \vee c_{si}, \dots, m_n \vee c_{sn}), \\ m_i, c_{si} \in \{0,1\}, s = \overline{1, p}\}, \quad (1.4)$$

$$mC = \{mc_s \mid mc_s = (m_1 c_{s1}, m_2 c_{s2}, \dots, m_i c_{si}, \dots, m_n c_{sn}), m_i, c_{si} \in \{0,1\}, s = \overline{1, p}\}, \quad (1.5)$$

при цьому  $|mC| = p$  – число рядків результуючої матриці.

### 1.1.1 Комутативний закон:

$$m \vee A = A \vee m; m \wedge A = A \wedge m. \quad (1.6)$$

### 1.1.2 Асоціативний закон:

$$(a \vee b) \vee C = a \vee (b \vee C), \quad (1.7)$$

$$(ab)C = a(bC), \quad (1.8)$$

де  $a = (a_1, a_2, \dots, a_i, \dots, a_n)$ ,  $b = (b_1, b_2, \dots, b_i, \dots, b_n)$ ,  $a_i, b_i \in \{0,1\}$ ,  $i = \overline{1, n}$ ,

$$C = \{c \mid c_s = (c_{s1}, c_{s2}, \dots, c_{si}, \dots, c_{sn}), c_{si} \in \{0,1\}, s = \overline{1, p}, i = \overline{1, n}\}.$$

Доведення. Диз'юнкцію (кон'юнкцію) векторів  $a$ ,  $b$  і матриці  $C$  для лівих частин рівностей (1.7), (1.8) можна зобразити у вигляді:

$$(a \vee b) \vee C = \left( \begin{array}{c} \left[ \begin{array}{c} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{array} \right] \vee \left[ \begin{array}{c} b_1 \\ b_2 \\ \dots \\ b_i \\ \dots \\ b_n \end{array} \right] \end{array} \right) \vee \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} =$$

$$= \begin{bmatrix} a_1 \vee b_1 \\ a_2 \vee b_2 \\ \dots \\ a_i \vee b_i \\ \dots \\ a_n \vee b_n \end{bmatrix} \vee \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} =$$

$$= \begin{bmatrix} (a_1 \vee b_1) \vee c_{11} & (a_2 \vee b_2) \vee c_{12} & \dots & (a_i \vee b_i) \vee c_{1i} & \dots & (a_n \vee b_n) \vee c_{1n} \\ (a_1 \vee b_1) \vee c_{21} & (a_2 \vee b_2) \vee c_{22} & \dots & (a_i \vee b_i) \vee c_{2i} & \dots & (a_n \vee b_n) \vee c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ (a_1 \vee b_1) \vee c_{s1} & (a_2 \vee b_2) \vee c_{s2} & \dots & (a_i \vee b_i) \vee c_{si} & \dots & (a_n \vee b_n) \vee c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ (a_1 \vee b_1) \vee c_{p1} & (a_2 \vee b_2) \vee c_{p2} & \dots & (a_i \vee b_i) \vee c_{pi} & \dots & (a_n \vee b_n) \vee c_{pn} \end{bmatrix}, \quad (1.9)$$

$$(ab)C = \begin{pmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_i \\ \dots \\ b_n \end{bmatrix} \end{pmatrix} \cdot \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} =$$

$$= \begin{bmatrix} a_1 b_1 \\ a_2 b_2 \\ \dots \\ a_i b_i \\ \dots \\ a_n b_n \end{bmatrix} \cdot \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} =$$

$$= \begin{bmatrix} (a_1 b_1) c_{11} & (a_2 b_2) c_{12} & \dots & (a_i b_i) c_{1i} & \dots & (a_n b_n) c_{1n} \\ (a_1 b_1) c_{21} & (a_2 b_2) c_{22} & \dots & (a_i b_i) c_{2i} & \dots & (a_n b_n) c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ (a_1 b_1) c_{s1} & (a_2 b_2) c_{s2} & \dots & (a_i b_i) c_{si} & \dots & (a_n b_n) c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ (a_1 b_1) c_{p1} & (a_2 b_2) c_{p2} & \dots & (a_i b_i) c_{pi} & \dots & (a_n b_n) c_{pn} \end{bmatrix}. \quad (1.10)$$

Відповідно, праві частини рівностей (1.7), (1.8) перетворюються до виду:



$$a \vee (b \vee C) = \begin{bmatrix} a_1 \vee (b_1 \vee c_{11}) & a_2 \vee (b_2 \vee c_{12}) & \dots & a_i \vee (b_i \vee c_{li}) & \dots & a_n \vee (b_n \vee c_{ln}) \\ a_1 \vee (b_1 \vee c_{21}) & a_2 \vee (b_2 \vee c_{22}) & \dots & a_i \vee (b_i \vee c_{2i}) & \dots & a_n \vee (b_n \vee c_{2n}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee (b_1 \vee c_{s1}) & a_2 \vee (b_2 \vee c_{s2}) & \dots & a_i \vee (b_i \vee c_{si}) & \dots & a_n \vee (b_n \vee c_{sn}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee (b_1 \vee c_{p1}) & a_2 \vee (b_2 \vee c_{p2}) & \dots & a_i \vee (b_i \vee c_{pi}) & \dots & a_n \vee (b_n \vee c_{pn}) \end{bmatrix}, \quad (1.11)$$

$$a(bC) = \begin{bmatrix} a_1(b_1c_{11}) & a_2(b_2c_{12}) & \dots & a_i(b_ic_{li}) & \dots & a_n(b_nc_{ln}) \\ a_1(b_1c_{21}) & a_2(b_2c_{22}) & \dots & a_i(b_ic_{2i}) & \dots & a_n(b_nc_{2n}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1(b_1c_{s1}) & a_2(b_2c_{s2}) & \dots & a_i(b_ic_{si}) & \dots & a_n(b_nc_{sn}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1(b_1c_{p1}) & a_2(b_2c_{p2}) & \dots & a_i(b_ic_{pi}) & \dots & a_n(b_nc_{pn}) \end{bmatrix}. \quad (1.12)$$

Справедливість асоціативного закону у формі (1.7), (1.8) впливає з таблиць істинності 1.1, 1.2, які доводять покомпонентну рівність відповідних матриць (1.9) і (1.11), (1.10) і (1.12):

Таблиця 1.1 – Доведення асоціативного закону (1.7) для диз'юнкції

$$(a_i \vee b_i) \vee c_{si} = a_i \vee (b_i \vee c_{si})$$

№	$a_i$	$b_i$	$c_{si}$	$a_i \vee b_i$	$(a_i \vee b_i) \vee c_{si}$	$b_i \vee c_{si}$	$a_i \vee (b_i \vee c_{si})$
0	0	0	0	0	0	0	0
1	0	0	1	0	1	1	1
2	0	1	0	1	1	1	1
3	0	1	1	1	1	1	1
4	1	0	0	1	1	0	1
5	1	0	1	1	1	1	1
6	1	1	0	1	1	1	1
7	1	1	1	1	1	1	1

Таблиця 1.2 – Доведення асоціативного закону (1.8) для кон'юнкції

$$(a_i b_i) c_{si} = a_i (b_i c_{si})$$

№	$a_i$	$b_i$	$c_{si}$	$a_i b_i$	$(a_i b_i) c_{si}$	$b_i c_{si}$	$a_i (b_i c_{si})$
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	1	0	0	0	0	0
3	0	1	1	0	0	1	0
4	1	0	0	0	0	0	0
5	1	0	1	0	0	0	0
6	1	1	0	1	0	0	0
7	1	1	1	1	1	1	1

### 1.1.3 Дистрибутивний закон:

$$a \vee bC = (a \vee b)(a \vee C), \quad (1.13)$$

$$a(b \vee C) = ab \vee aC. \quad (1.14)$$

Доведення. У дистрибутивному законі беруть участь три компоненти, з них – два  $n$ -мірних двійкових вектори  $a$ ,  $b$ , і сукупність  $C$   $n$ -мірних двійкових векторів:

$$a = (a_1, a_2, \dots, a_i, \dots, a_n), \quad b = (b_1, b_2, \dots, b_i, \dots, b_n), \quad a_i, b_i \in \{0,1\}, \quad i = \overline{1, n},$$

$$C = \{c_s \mid c_s = (c_{s1}, c_{s2}, \dots, c_{si}, \dots, c_{sn}), c_{si} \in \{0,1\}, s = \overline{1, p}, i = \overline{1, n}\}, \quad |C| = p.$$

Перевірка виконання рівності (1.12)  $a \vee bC = (a \vee b)(a \vee C)$  дає наступне:

$$\begin{aligned}
a \vee bC &= \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \vee \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_i \\ \dots \\ b_n \end{bmatrix} \cdot \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} = \\
&= \left( \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \vee \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_i \\ \dots \\ b_n \end{bmatrix} \right) \cdot \left( \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \vee \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} \right) = (a \vee b)(a \vee C), \quad (1.15)
\end{aligned}$$

$$\begin{aligned}
&= \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \vee \begin{bmatrix} b_1c_{11} & b_2c_{12} & \dots & b_1c_{1i} & \dots & b_n c_{1n} \\ b_1c_{21} & b_2c_{22} & \dots & b_1c_{2i} & \dots & b_n c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_1c_{s1} & b_2c_{s2} & \dots & b_1c_{si} & \dots & b_n c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_1c_{p1} & b_2c_{p2} & \dots & b_1c_{pi} & \dots & b_n c_{pn} \end{bmatrix} = \\
&= \begin{bmatrix} a_1 \vee b_1 \\ a_2 \vee b_2 \\ \dots \\ a_i \vee b_i \\ \dots \\ a_n \vee b_n \end{bmatrix} \cdot \begin{bmatrix} a_1 \vee c_{11} & a_2 \vee c_{12} & \dots & a_i \vee c_{1i} & \dots & a_n \vee c_{1n} \\ a_1 \vee c_{21} & a_2 \vee c_{22} & \dots & a_i \vee c_{2i} & \dots & a_n \vee c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee c_{s1} & a_2 \vee c_{s2} & \dots & a_i \vee c_{si} & \dots & a_n \vee c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee c_{p1} & a_2 \vee c_{p2} & \dots & a_i \vee c_{pi} & \dots & a_n \vee c_{pn} \end{bmatrix}, \quad (1.16)
\end{aligned}$$

$$\begin{aligned}
&= \begin{bmatrix} a_1 \vee b_1c_{11} & a_2 \vee b_2c_{12} & \dots & a_i \vee b_1c_{1i} & \dots & a_n \vee b_n c_{1n} \\ a_1 \vee b_1c_{21} & a_2 \vee b_2c_{22} & \dots & a_i \vee b_1c_{2i} & \dots & a_n \vee b_n c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee b_1c_{s1} & a_2 \vee b_2c_{s2} & \dots & a_i \vee b_1c_{si} & \dots & a_n \vee b_n c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee b_1c_{p1} & a_2 \vee b_2c_{p2} & \dots & a_i \vee b_1c_{pi} & \dots & a_n \vee b_n c_{pn} \end{bmatrix} =
\end{aligned}$$

$$= \begin{bmatrix} (a_1 \vee b_1)(a_1 \vee c_{11}) & (a_2 \vee b_2)(a_2 \vee c_{12}) & \dots & (a_i \vee b_i)(a_i \vee c_{1i}) & \dots & (a_n \vee b_n)(a_n \vee c_{1n}) \\ (a_1 \vee b_1)(a_1 \vee c_{21}) & (a_2 \vee b_2)(a_2 \vee c_{22}) & \dots & (a_i \vee b_i)(a_i \vee c_{2i}) & \dots & (a_n \vee b_n)(a_n \vee c_{2n}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ (a_1 \vee b_1)(a_1 \vee c_{s1}) & (a_2 \vee b_2)(a_2 \vee c_{s2}) & \dots & (a_i \vee b_i)(a_i \vee c_{si}) & \dots & (a_n \vee b_n)(a_n \vee c_{sn}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ (a_1 \vee b_1)(a_1 \vee c_{p1}) & (a_2 \vee b_2)(a_2 \vee c_{p2}) & \dots & (a_i \vee b_i)(a_i \vee c_{pi}) & \dots & (a_n \vee b_n)(a_n \vee c_{pn}) \end{bmatrix}, \quad (1.17)$$

Формула (1.14)  $a(b \vee C) = ab \vee aC$  при підстановці векторів і матриць має вигляд:

$$\begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \cdot \left( \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_i \\ \dots \\ b_n \end{bmatrix} \vee \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} \right) =$$

$$= \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_i \\ \dots \\ b_n \end{bmatrix} \vee \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \cdot \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix}, \quad (1.18)$$

$$\begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \cdot \begin{bmatrix} b_1 \vee c_{11} & b_2 \vee c_{12} & \dots & b_i \vee c_{1i} & \dots & b_n \vee c_{1n} \\ b_1 \vee c_{21} & b_2 \vee c_{22} & \dots & b_i \vee c_{2i} & \dots & b_n \vee c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_1 \vee c_{s1} & b_2 \vee c_{s2} & \dots & b_i \vee c_{si} & \dots & b_n \vee c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_1 \vee c_{p1} & b_2 \vee c_{p2} & \dots & b_i \vee c_{pi} & \dots & b_n \vee c_{pn} \end{bmatrix} =$$

$$= \begin{bmatrix} a_1 b_1 \\ a_2 b_2 \\ \dots \\ a_i b_i \\ \dots \\ a_n b_n \end{bmatrix} \vee \begin{bmatrix} a_1 c_{11} & a_2 c_{12} & \dots & a_i c_{1i} & \dots & a_n c_{1n} \\ a_1 c_{21} & a_2 c_{22} & \dots & a_i c_{2i} & \dots & a_n c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 c_{s1} & a_2 c_{s2} & \dots & a_i c_{si} & \dots & a_n c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 c_{p1} & a_2 c_{p2} & \dots & a_i c_{pi} & \dots & a_n c_{pn} \end{bmatrix}, \quad (1.19)$$

$$\begin{bmatrix} a_1(b_1 \vee c_{11}) & a_2(b_2 \vee c_{12}) & \dots & a_i(b_i \vee c_{1i}) & \dots & a_n(b_n \vee c_{1n}) \\ a_1(b_1 \vee c_{21}) & a_2(b_2 \vee c_{22}) & \dots & a_i(b_i \vee c_{2i}) & \dots & a_n(b_n \vee c_{2n}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1(b_1 \vee c_{s1}) & a_2(b_2 \vee c_{s2}) & \dots & a_i(b_i \vee c_{si}) & \dots & a_n(b_n \vee c_{sn}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1(b_1 \vee c_{p1}) & a_2(b_2 \vee c_{p2}) & \dots & a_i(b_i \vee c_{pi}) & \dots & a_n(b_n \vee c_{pn}) \end{bmatrix} =$$

$$= \begin{bmatrix} a_1 b_1 \vee a_1 c_{11} & a_2 b_2 \vee a_2 c_{12} & \dots & a_i b_i \vee a_i c_{1i} & \dots & a_n b_n \vee a_n c_{1n} \\ a_1 b_1 \vee a_1 c_{21} & a_2 b_2 \vee a_2 c_{22} & \dots & a_i b_i \vee a_i c_{2i} & \dots & a_n b_n \vee a_n c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 b_1 \vee a_1 c_{s1} & a_2 b_2 \vee a_2 c_{s2} & \dots & a_i b_i \vee a_i c_{si} & \dots & a_n b_n \vee a_n c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 b_1 \vee a_1 c_{p1} & a_2 b_2 \vee a_2 c_{p2} & \dots & a_i b_i \vee a_i c_{pi} & \dots & a_n b_n \vee a_n c_{pn} \end{bmatrix}. \quad (1.20)$$

Поелементна дорівняність матриць у формулах (1.17), (1.20) повинне виконуватися таким чином:

$$a_i \vee b_i c_{si} = (a_i \vee b_i)(a_i \vee c_{si}), \quad (1.21)$$

$$a_i(b_i \vee c_{si}) = a_i b_i \vee a_i c_{si}, \quad (1.22)$$

де  $i = \overline{1, n}, s = \overline{1, p}$ . Область визначення для рівностей (1.21), (1.22) обмежується стандартними  $2^3 = 8$  двійковими наборами. Доведення подається у вигляді таблиць істинності 1.3, 1.4:

Таблиця 1.3 – Доведення дистрибутивного закону у формі (1.21)

$$a_i \vee b_i c_{si} = (a_i \vee b_i)(a_i \vee c_{si}), i = \overline{1, n}, s = \overline{1, p}$$

№	$a_i$	$b_i$	$c_{si}$	$b_i c_{si}$	$a_i \vee b_i c_{si}$	$a_i \vee b_i$	$a_i \vee c_{si}$	$(a_i \vee b_i)(a_i \vee c_{si})$
0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0
2	0	1	0	0	0	1	0	0
3	0	1	1	1	1	1	1	1
4	1	0	0	0	1	1	1	1
5	1	0	1	0	1	1	1	1
6	1	1	0	0	1	1	1	1
7	1	1	1	1	1	1	1	1

Таблиця 1.4 – Покомпонентне виконання дистрибутивного закону (1.22)

$$a_i(b_i \vee c_{si}) = a_i b_i \vee a_i c_{si}, i = \overline{1, n}, s = \overline{1, p}$$

№	$a_i$	$b_i$	$c_{si}$	$b_i \vee c_{si}$	$a_i(b_i \vee c_{si})$	$a_i b_i$	$a_i c_{si}$	$a_i b_i \vee a_i c_{si}$
0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0
2	0	1	0	1	0	0	0	0
3	0	1	1	1	0	0	0	0
4	1	0	0	0	0	0	0	0
5	1	0	1	1	1	0	1	1
6	1	1	0	1	1	1	0	1
7	1	1	1	1	1	1	1	1

Можна показати, що дистрибутивний закон у формі

$$ab \vee C \neq (a \vee C)(b \vee C) \quad (1.23)$$

не виконується у векторно-матричній алгебрі логіки, оскільки не визначене множення матриць, що фігурує в правій частині (1.23):

$$\begin{aligned} & \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_i \\ \dots \\ b_n \end{bmatrix} \vee \begin{bmatrix} c_{11} & c_{12} & \dots & c_{li} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} \neq \\ & \left( \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \vee \begin{bmatrix} c_{11} & c_{12} & \dots & c_{li} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} \right) \cdot \left( \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_i \\ \dots \\ b_n \end{bmatrix} \vee \begin{bmatrix} c_{11} & c_{12} & \dots & c_{li} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} \right), \\ & \tag{1.24} \end{aligned}$$

$$\begin{aligned} & \begin{bmatrix} a_1 b_1 \\ a_2 b_2 \\ \dots \\ a_i b_i \\ \dots \\ a_n b_n \end{bmatrix} \vee \begin{bmatrix} c_{11} & c_{12} & \dots & c_{li} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} \neq \\ & \neq \begin{bmatrix} a_1 \vee c_{11} & a_2 \vee c_{12} & \dots & a_i \vee c_{li} & \dots & a_n \vee c_{1n} \\ a_1 \vee c_{21} & a_2 \vee c_{22} & \dots & a_i \vee c_{2i} & \dots & a_n \vee c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee c_{s1} & a_2 \vee c_{s2} & \dots & a_i \vee c_{si} & \dots & a_n \vee c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee c_{p1} & a_2 \vee c_{p2} & \dots & a_i \vee c_{pi} & \dots & a_n \vee c_{pn} \end{bmatrix}. \end{aligned}$$

$$\begin{bmatrix} b_1 \vee c_{11} & b_2 \vee c_{12} & \dots & b_i \vee c_{1i} & \dots & b_n \vee c_{1n} \\ b_1 \vee c_{21} & b_2 \vee c_{22} & \dots & b_i \vee c_{2i} & \dots & b_n \vee c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_1 \vee c_{s1} & b_2 \vee c_{s2} & \dots & b_i \vee c_{si} & \dots & b_n \vee c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_1 \vee c_{p1} & b_2 \vee c_{p2} & \dots & b_i \vee c_{pi} & \dots & b_n \vee c_{pn} \end{bmatrix}, \quad (1.25)$$

Якщо ввести правило поелементного множення двійкових матриць однакової розмірності, то можна розглядати можливість виконання дистрибутивного закону у формі (1.23).

Аналогічно можна показати, що не виконується дистрибутивний закон у формі:

$$(a \vee b)C \neq aC \vee bC, \quad (1.26)$$

оскільки не визначене додавання матриць.

1.1.4 Дії з константами. Константами є  $n$ -мірні нульовий і одиничний вектори:  $m^0 = (0,0,\dots,0)$ ,  $m^1 = (1,1,\dots,1)$ .

$$m \vee m^0 = m, \quad m \cdot m^0 = m^0, \quad (1.27)$$

$$m \vee m^1 = m^1, \quad m \cdot m^1 = m, \quad (1.28)$$

$$\overline{m^0} = m^1, \quad \overline{m^1} = m^0. \quad (1.29)$$

$$m^1 \cdot C = C, \quad m^0 \vee C = C, \quad (1.30)$$

$$m^0 \cdot C = m^0, \quad m^1 \vee C = m^1. \quad (1.31)$$



Дії (1.31) використовуються при доведенні законів елімінації й склеювання, що буде показано нижче.

1.1.5 Закон протиріччя (для векторів):

$$m \cdot \bar{m} = m^0. \quad (1.32)$$

1.1.6 Закон виключеного третього (для векторів):

$$m \vee \bar{m} = m^1. \quad (1.33)$$

1.1.7 Закон елімінації (поглинання):

$$a(a \vee C) = a, \quad a \vee aC = a. \quad (1.34)$$

Доведення. Виконання еквівалентних перетворень на основі застосування операцій з константами й дистрибутивним законом дозволяє одержати:

$$a(a \vee C) \stackrel{(1.27)}{=} (a \vee m^0)(a \vee C) \stackrel{(1.13)}{=} a \vee m^0 C \stackrel{(1.26)}{=} a \vee m^0 \stackrel{(1.27)}{=} a,$$

$$a \vee aC \stackrel{(1.28)}{=} am^1 \vee aC \stackrel{(1.14)}{=} a(m^1 \vee C) \stackrel{(1.21)}{=} am^1 \stackrel{(1.28)}{=} a.$$

1.1.8 Закон склеювання. Виконується в наступних формулюваннях:

$$m\bar{m} \vee C = C, \quad (1.35)$$

$$(m \vee \bar{m})C = C. \quad (1.36)$$

Доведення. Варто помітити, що ліві частини законів склеювання (1.35), (1.36) не можуть бути розкладені за дистрибутивним законом, оскільки відповідні його формулювання не визначені у векторно-матричній алгебрі. Для

доведення (1.35), (1.36) застосування законів виключеного третього, протиріччя й дій з константами виявляється достатнім:

$$m\bar{m} \vee C \stackrel{(1.27)}{=} m^0 \vee C \stackrel{(1.21)}{=} C,$$

$$(m \vee \bar{m})C \stackrel{(1.33)}{=} m^1 C \stackrel{(1.31)}{=} C.$$

Слід також звернути увагу, що закон елімінації у формах  $(C \vee b)(C \vee \bar{b}) = C$  і  $Cb \vee C\bar{b} = C$  не виконується, оскільки у векторно-матричній алгебрі визначене перемножування вектора й матриці, але не визначене логічне додавання й множення матриць.

#### 1.1.9 Закон Блейка-Порецького:

$$a(\bar{a} \vee C) = aC, \quad a \vee \bar{a}C = a \vee C, \quad (1.37)$$

$$\bar{a}(a \vee C) = \bar{a}C, \quad \bar{a} \vee aC = \bar{a} \vee C. \quad (1.38)$$

Доведення. Необхідно скористатися дистрибутивним законом. Тоді на прикладі першої рівності (1.37) можна одержати:

$$a(\bar{a} \vee C) \stackrel{(1.14)}{=} a\bar{a} \vee aC \stackrel{(1.32)}{=} m^0 \vee aC \stackrel{(1.31)}{=} aC.$$

Інші формулювання закону Порецького доводяться аналогічно.

1.1.10 Закон ідемпотентності (тільки для векторів, але не для матриць, оскільки логічні додавання й множення матриць не визначені):

$$m \vee m = m, \quad m \cdot m = m. \quad (1.39)$$

## 1.1.11 Закон інволюції:

$$\overline{\overline{m}} = m, \quad \overline{\overline{C}} = C. \quad (1.40)$$

## 1.1.12 Закон Де Моргана:

$$\overline{a \vee C} = \bar{a} \cdot \bar{C}, \quad (1.41)$$

$$\overline{aC} = \bar{a} \vee \bar{C}. \quad (1.42)$$

Доведення. На підставі покомпонентних операцій над елементами векторів і матриць справедливність формул (1.41), (1.42) можна показати за допомогою таблиць істинності. В (1.41), (1.42) беруть участь два компоненти:  $n$ -мірний вектор і сукупність  $n$ -мірних векторів у вигляді матриці, підстановка компонентів яких в (1.41) дає:

$$\begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \vee \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_i \\ \dots \\ a_n \end{bmatrix} \cdot \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1i} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2i} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{si} & \dots & c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p1} & c_{p2} & \dots & c_{pi} & \dots & c_{pn} \end{bmatrix},$$

$$\begin{bmatrix} a_1 \vee c_{11} & a_2 \vee c_{12} & \dots & a_i \vee c_{1i} & \dots & a_n \vee c_{1n} \\ a_1 \vee c_{21} & a_2 \vee c_{22} & \dots & a_i \vee c_{2i} & \dots & a_n \vee c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee c_{s1} & a_2 \vee c_{s2} & \dots & a_i \vee c_{si} & \dots & a_n \vee c_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 \vee c_{p1} & a_2 \vee c_{p2} & \dots & a_i \vee c_{pi} & \dots & a_n \vee c_{pn} \end{bmatrix} = \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \dots \\ \bar{a}_i \\ \dots \\ \bar{a}_n \end{bmatrix} \cdot \begin{bmatrix} \bar{c}_{11} & \bar{c}_{12} & \dots & \bar{c}_{1i} & \dots & \bar{c}_{1n} \\ \bar{c}_{21} & \bar{c}_{22} & \dots & \bar{c}_{2i} & \dots & \bar{c}_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \bar{c}_{s1} & \bar{c}_{s2} & \dots & \bar{c}_{si} & \dots & \bar{c}_{sn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \bar{c}_{p1} & \bar{c}_{p2} & \dots & \bar{c}_{pi} & \dots & \bar{c}_{pn} \end{bmatrix},$$

$$\begin{aligned}
& \left[ \begin{array}{cccccc}
\overline{a_1 \vee c_{11}} & \overline{a_2 \vee c_{12}} & \dots & \overline{a_i \vee c_{1i}} & \dots & \overline{a_n \vee c_{1n}} \\
\overline{a_1 \vee c_{21}} & \overline{a_2 \vee c_{22}} & \dots & \overline{a_i \vee c_{2i}} & \dots & \overline{a_n \vee c_{2n}} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
\overline{a_1 \vee c_{s1}} & \overline{a_2 \vee c_{s2}} & \dots & \overline{a_i \vee c_{si}} & \dots & \overline{a_n \vee c_{sn}} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
\overline{a_1 \vee c_{p1}} & \overline{a_2 \vee c_{p2}} & \dots & \overline{a_i \vee c_{pi}} & \dots & \overline{a_n \vee c_{pn}}
\end{array} \right] = \\
& = \left[ \begin{array}{cccccc}
\overline{\bar{a}_1 \bar{c}_{11}} & \overline{\bar{a}_2 \bar{c}_{12}} & \dots & \overline{\bar{a}_i \bar{c}_{1i}} & \dots & \overline{\bar{a}_n \bar{c}_{1n}} \\
\overline{\bar{a}_1 \bar{c}_{21}} & \overline{\bar{a}_2 \bar{c}_{22}} & \dots & \overline{\bar{a}_i \bar{c}_{2i}} & \dots & \overline{\bar{a}_n \bar{c}_{2n}} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
\overline{\bar{a}_1 \bar{c}_{s1}} & \overline{\bar{a}_2 \bar{c}_{s2}} & \dots & \overline{\bar{a}_i \bar{c}_{si}} & \dots & \overline{\bar{a}_n \bar{c}_{sn}} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
\overline{\bar{a}_1 \bar{c}_{p1}} & \overline{\bar{a}_2 \bar{c}_{p2}} & \dots & \overline{\bar{a}_i \bar{c}_{pi}} & \dots & \overline{\bar{a}_n \bar{c}_{pn}}
\end{array} \right] \cdot \quad (1.43)
\end{aligned}$$

Для формули (1.42) перетворення виконуються аналогічно й дають:

$$\begin{aligned}
& \left[ \begin{array}{cccccc}
\overline{a_1 c_{11}} & \overline{a_2 c_{12}} & \dots & \overline{a_i c_{1i}} & \dots & \overline{a_n c_{1n}} \\
\overline{a_1 c_{21}} & \overline{a_2 c_{22}} & \dots & \overline{a_i c_{2i}} & \dots & \overline{a_n c_{2n}} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
\overline{a_1 c_{s1}} & \overline{a_2 c_{s2}} & \dots & \overline{a_i c_{si}} & \dots & \overline{a_n c_{sn}} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
\overline{a_1 c_{p1}} & \overline{a_2 c_{p2}} & \dots & \overline{a_i c_{pi}} & \dots & \overline{a_n c_{pn}}
\end{array} \right] = \\
& = \left[ \begin{array}{cccccc}
\overline{\bar{a}_1 \vee \bar{c}_{11}} & \overline{\bar{a}_2 \vee \bar{c}_{12}} & \dots & \overline{\bar{a}_i \vee \bar{c}_{1i}} & \dots & \overline{\bar{a}_n \vee \bar{c}_{1n}} \\
\overline{\bar{a}_1 \vee \bar{c}_{21}} & \overline{\bar{a}_2 \vee \bar{c}_{22}} & \dots & \overline{\bar{a}_i \vee \bar{c}_{2i}} & \dots & \overline{\bar{a}_n \vee \bar{c}_{2n}} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
\overline{\bar{a}_1 \vee \bar{c}_{s1}} & \overline{\bar{a}_2 \vee \bar{c}_{s2}} & \dots & \overline{\bar{a}_i \vee \bar{c}_{si}} & \dots & \overline{\bar{a}_n \vee \bar{c}_{sn}} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
\overline{\bar{a}_1 \vee \bar{c}_{p1}} & \overline{\bar{a}_2 \vee \bar{c}_{p2}} & \dots & \overline{\bar{a}_i \vee \bar{c}_{pi}} & \dots & \overline{\bar{a}_n \vee \bar{c}_{pn}}
\end{array} \right] \cdot \quad (1.44)
\end{aligned}$$

Таблиці істинності, побудовані для рівностей (1.43) і (1.44), будуть включати  $2^2 = 4$  стандартних двійкових наборів, для яких виконуються відповідні покомпонентні операції (табл. 1.5, 1.6).

Таблиця 1.5 – Перевірка покомпонентної рівності матриць для (1.43)

$$\overline{a_i \vee c_{si}} = \bar{a}_i \cdot \bar{c}_{si}, i = \overline{1, n}, s = \overline{1, p}$$

N	$a_i$	$c_{si}$	$a_i \vee c_{si}$	$\overline{a_i \vee c_{si}}$	$\bar{a}_i$	$\bar{c}_{si}$	$\bar{a}_i \cdot \bar{c}_{si}$
0	0	0	0	1	1	1	1
1	0	1	1	0	1	0	0
2	1	0	1	0	0	1	0
3	1	1	1	0	0	0	0

Таблиця 1.6 – Перевірка покомпонентної рівності матриць для закону Де Моргана у формі (1.44)

$$\overline{a_i c_{si}} = \bar{a}_i \vee \bar{c}_{si}, i = \overline{1, n}, s = \overline{1, p}$$

N	$a_i$	$b_i$	$a_i c_{si}$	$\overline{a_i c_{si}}$	$\bar{a}_i$	$\bar{c}_{si}$	$\bar{a}_i \vee \bar{c}_{si}$
0	0	0	0	1	1	1	1
1	0	1	0	1	1	0	1
2	1	0	0	1	0	1	1
3	1	1	1	0	0	0	0

Таким чином, закон Де Моргана для векторно-матричної логіки доведений.

## 1.2 Висновки

1. Наукова новизна. Запропоновано нову векторно-матричну алгебологічну систему, призначену для використання в інфраструктурі векторного логічного аналізу, заснованого на застосуванні мультипроцесорного мозгоподібного комп'ютера.

2. Векторно-матрична алгебра логіки, де сигнатура задає стандартний набір логічних операцій *and*, *or*, *not* для визначення взаємодії між сумісними елементами з носія, що утворюють двійкові  $n$ -мірні вектори й двійкові матриці розмірністю  $p \times n$ . Операція диз'юнкції (кон'юнкції) вектора й матриці виконується поелементно, відповідно до введених правил. Виключено операції між матрицями, що обмежує формулювання деяких законів і тотожностей векторно-матричної логіки, аналогічних властивостям булевої алгебри. Векторно-матрична алгебра може бути розширена у випадку її доповнення логічним множенням і додаванням матриць, що є перспективним дослідженням у майбутньому.

3. Практична значимість запропонованих результатів досліджень полягає в орієнтації введеної векторно-матричної алгебри логіки на ефективний аналіз векторних логічних процедур за допомогою мозгоподібних обчислювальних структур.

## 2 ВЕКТОРНИЙ ЛОГІЧНИЙ АНАЛІЗ ДІАГНОСТИЧНОЇ ІНФОРМАЦІЇ

Пропонується інфраструктура асоціативно-логічного аналізу інформації на основі швидкодіючого мультипроцесора для пошуку, розпізнавання й прийняття рішень в  $n$ -мірному векторному дискретному просторі [101, 103, 106, 115-117, 119, 121]. Представлено алгебраїчну структуру, що регулює векторно-матричні неарифметичні перетворення в дискретному булевому просторі для аналізу інформації на основі логічних операцій над асоціативними даними. Розглядається застосування алгебри векторної логіки [116] для створення процес-моделей актуальних прикладних завдань, якість розв'язок яких оцінюється примітивною інтегральною неарифметичною метрикою взаємодії асоціативних структур.

### 2.1 Інфраструктура формування умов та прийняття рішень

Асоціативно-логічний комп'ютер – новий спеціалізований движок для Internet простору. Інфраструктура глобальної мережі Internet по оцінці Gary Smith (Gartner Research Group) має потребу в радикальній перебудові. Глобальний інформаційний простір уже не вміщається в демократично розвиваються й тому недосконалі формати й структури даних. Ще гірше справа з розробкою движків, спеціалізованих під завдання швидкого пересування по просторі для пошуку, розпізнавання інформації й прийняття рішень. Роль таких пристроїв сьогодні виконують універсальні процесори, функціональність яких на 90% не придатна для розв'язку згаданих задач. На сьогодні є неструктурований і нестандартизований глобальний інформаційний простір, практично відсутність спеціалізованих під Internet процесорів. Виключення може бути представлено першим виробом у вигляді iPad планшета, процесор якого виконаний у вигляді системи на кристалі Apple A4, що складається із багатоядерного ARM-процесора Cortex-A9 MPCore, контролерів пам'яті й

графіки, заточеними під Internet, що підтверджує актуальність створення нових движків і величезну їхню ринкову привабливість сьогодні й у майбутньому. У зв'язку із цим цікаво згадати й основні закони розвитку інфраструктури інформаційного простору: 1) Закон Мура – число транзисторів на кристалі в мікросхемах щорічно подвоюється і їхня кількість зростає швидше, ніж необхідний обсяг для проектування складних цифрових систем. 2) Закон Рока – вартість устаткування для випуску напівпровідникових кристалів подвоюється кожні 4 роки. Інвестиції в технологічний процес переважають над капіталовкладеннями в створення нових цифрових і мікропроцесорних систем. 3) Закон Макроуна – комп'ютер, що ви хочете купити, завжди коштує 1000 доларів. 4) Закон Меткафа – ріст значимості мережі визначається квадратом числа її користувачів. 5) Закон Вірта – зниження швидкості програмного забезпечення відбувається швидше росту продуктивності апаратури. Витрати на розробку програмного забезпечення комп'ютерної системи становлять 80% від її вартості. 6) Витрати на тестування мають експонентну залежність від обсягу випуску продукції. Число інженерів-верифікаторів в 2 рази перевищує кількість проектувальників програмних і апаратних продуктів. Висновок: завжди буде мало продуктивності, пам'яті, необхідних сервісів, комп'ютер завжди буде дорого коштувати, користувач завжди буде оплачувати зайві функціональності.

Що можна зробити для розв'язку зазначених вище проблем? Далі представлені сім параметрів підвищення продуктивності мозгоподібного комп'ютера, які істотно впливають на інфраструктуру (рис. 2.1) аналізу даних для ухвалення розв'язок.



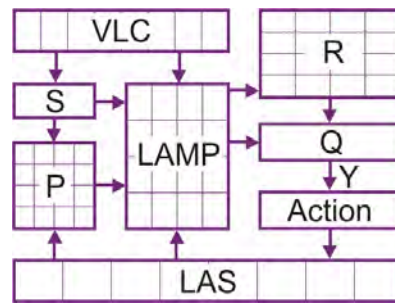


Рисунок 2.1 – Інфраструктура логічного ухвалення розв’язок

1) Виключення арифметичних операцій із системи команд процесора. 2) Використання в обчислювальних процедурах алгебри векторної логіки. Комп’ютер виграє в людини в здатності швидко аналізувати велике число істотних логічних змінних, зконкатенованих у вектор-запит. 3) Застосування мультипроцесорної матриці для розпаралелювання обчислювальних процедур. 4) Використання векторного двійкового критерію якості для оцінювання розв’язку. 5) Введення операції векторизації з метою генерування асоціації мінімального S-вектора істотних змінних, необхідних для пошуку розв’язку. 6) Створення вектор-біт D-оператора девекторизації для формування двійкового розв’язку на основі застосування логічних операцій *and*, *or*, *not* або їхньої комбінації до вектора (векторам) істотних змінних. 7) Створення P-платформи логічних процесів-моделей (IP-cores) для пошуку, розпізнавання й прийняття рішень.

Для розв’язку конкретного завдання, специфікованого S-вектором, необхідно знайти (квазі-) оптимальне покриття специфікації функціональностями, що формують P-платформу у вигляді сукупності IP-cores, згенерованих раніше. При відсутності конкретної процес-моделі в бібліотеці рішень P її слід синтезувати на основі аналізу логічного асоціативного простору L, позначеного на рис. 2.1 як LAS (Logical Associative Space). Фактично, такий простір являє собою Internet, бази даних і знань, засобу (програмного й апаратного) синтезу й аналізу інформації, що представлена у вигляді логічних асоціацій. Розв’язок завдання покриття функцій специфікації примітивними

процес-моделями платформи ґрунтується на використанні Y-технології [19], відомої також за назвою ESL – Electronic System Level design. Всі локальні й глобальні розв’язки оцінюються системою векторно-логічних критеріїв (VLC – Vector Logical Criteria). Дійсно, кожна процес-модель з P-платформи ставиться у відповідність критерій якості з банку VLC.

Інтегральна аналітична процес-модель аналізу інформації, що відповідає структурам даних, представленим на рис. 2.1, має вигляд:

$$\left\{ \begin{array}{l} R = [\max_{j=1}^m (S \wedge P_j)] \Delta_{i=1}^n L_i \leftarrow \max Q_i; \\ Q = \max Q_i \{ [\max_{j=1}^m (S \wedge P_j)] \Delta_{i=1}^n L_i \}; \\ Y = D[\max Q_i \{ [\max_{j=1}^m (S \wedge P_j)] \Delta_{i=1}^n L_i \}]. \end{array} \right.$$

Тут  $P_j \in P$ ,  $\Delta = \{\text{and, or, not, xor, slc}\}$  – векторні логічні операції, де slc – shift left bit crowding;  $Y = \{0, 1\}$  – двійковий розв’язок, спрямований на виконання конкретної активності (action);  $Q$  – критерій якості розв’язку, обумовлений двійковим вектором;  $L_i$  – варіант змістовного розв’язку, представленого у вигляді інформаційного повідомлення;  $m$  – кількість процес-моделей у платформі  $P$ ;  $n$  – число варіантів асоціацій, серед яких здійснюється пошук розв’язку;  $R$  – (квазі-)оптимальний змістовний розв’язок, представлений у вигляді сукупності векторів або інших інформаційних повідомлень у форматі синтаксису природних або штучних мов.

Далі пропонується алгебраїчна структура, що регулює векторно-матричні перетворення в дискретному булевому просторі для аналізу інформації (тестування й пошук дефектів) на основі логічних операцій над асоціативними даними.

Мета – істотне зменшення часу аналізу асоціативних структур даних за рахунок введення математичного апарату – алгебри векторної логіки для паралельного виконання векторних операцій на спеціалізованому мультипроцесорному пристрої.

Завдання: 1. Розробка сигнатури, що задовольняє системі аксіом, тотожностей і законів, для носія, що представлений сукупністю асоціативних векторів однакової довжини в логічному векторному просторі. 2. Розробка сигнатури взаємин для носія, представленого парою: асоціативний вектор – асоціативна матриця. 3. Розробка сигнатури перетворень для носія, представленого парою асоціативних матриць однакової розмірності.

Об'єкт дослідження – алгебраїчні структури й логічні простори, орієнтовані на створення математичної основи ефективних паралельних обчислювальних процесів, реалізованих у мультипроцесорному спеціалізованому виробі.

Предмет дослідження – алгебра й простір векторної логіки, представлена структурою (носій – сигнатура) і орієнтована на паралельні обчислювальні операції в дискретному просторі.

Джерела: 1. Технології паралельних обчислень на спеціалізованих мультипроцесорних системах [57, 58, 61, 62-64, 67-69]. 2. Алгебраїчні структури, орієнтовані на створення математичного апарату для паралельних обчислювальних процесів [59, 60, 63-66, 69]. 3. Процес-моделі розв'язку практичних завдань на основі ефективних паралельних обчислювальних процесів [61, 62, 11, 13, 67].

## 2.2 Процес-моделі аналізу асоціативних структур даних

Мозгоподібність (функціональна) припускає виключення арифметики й всіх нелогічних критеріїв (імовірність, функція приналежності). Мозок не знає арифметики, крім нелогічної таблиці множення, але щомиті (щохвилини) у

реальному масштабі часу приймає логічно обґрунтовані розв'язки. Вектор є асоціація. Алгебра асоціативної логіки є алгебра векторної логіки, у якій задані операції: and, or, not. Векторизація є конкатенація змінних. Девекторизація є автоматна, у загальному випадку, процедура формування двійкового розв'язку на основі певних двійкових значень вектора (векторів) істотних змінних. Процес-модель девекторизації логічних умов, підготовлених для ухвалення розв'язку, має дві альтернативні структури. Перша – комбінаційна, формує миттєво двійковий розв'язок, наприклад, при наявності одиничних значень на  $n$  входах елемента and. Друга – послідовносна або автоматна, – накопичує сукупність одиничних умов  $n$  змінних у часі за допомогою, наприклад, рахункової структури. Можлива комбінація у вигляді паралельно-послідовної або послідовно-паралельної схеми для прийняття рішення. У загальному випадку, девекторизатор, як цифровий формувач розв'язку, може бути представлений у вигляді автоматної моделі першого роду:

$$\begin{aligned} Y(t) &= f[m(t), Z(t-1)]; \\ Z(t) &= g[m(t), Z(t-1)]. \end{aligned}$$

Тут  $Y, Z, m$  – змінні виходу, стану й входів автомата девекторизації логічних значень вхідного вектора.

Автомат вироджується в тривіальний або комбінаційний, якщо значення виходу девекторизатора залежить тільки від вхідного вектора:

$$Y(t) = f[m(t)] \approx Y = f(m_1, m_2, \dots, m_i, \dots, m_n).$$

Девекторизатор комбінаційного типу може бути простим або складним. Простий реалізований вектор-біт операція and, or, and-not, or-not:

$$Y = \Delta(m) = (m_1 \Delta m_2 \Delta \dots \Delta m_i \Delta \dots \Delta m_n), \Delta = \{\text{and, or, and-not, or-not}\}.$$

Складні конструкції, що використовують логічну взаємодію двох векторів, представлені виразами:

$$Y = \{(a \wedge b); (a \vee b); a \wedge \overline{(a \wedge b)}; b \wedge \overline{(a \wedge b)}; (a \vee b) \wedge \overline{(a \wedge b)}\}.$$

Кожному варіанту взаємодії можна поставити у відповідність теоретико-множинну діаграму, а також логічну схему, що відповідає процес-моделі прийняття рішення (рис. 2.2).

$Y = (a \wedge b)$		
$Y = (a \vee b)$		
$Y = a \wedge \overline{(a \wedge b)}$		
$Y = b \wedge \overline{(a \wedge b)}$		
$Y = (a \vee b) \wedge \overline{(a \wedge b)}$		

Рисунок 2.2 – Примітивні процес-моделі прийняття рішень

Як правило, процедури прийняття критичних і відповідальних рішень визначені в просторі й у часі, що виключає помилку, але при цьому затримується процес формування виводу. Для прийняття рішення конструктивний мозок синтезує тільки мінімальне число істотних змінних, крім несуттєвого придбання досвіду. Розв'язок завжди двійковий, отже, змінні асоціативного вектора, навіть лінгвістичні, що беруть участь у формуванні виводу, повинні бути наведені до двійкової норми.

На ринку електронних технологій найпоширенішими є дві альтернативні моделі даних: явна таблична й неявна аналітична. Таблиця істинності є сукупність векторів, що задають поведінку дискретного об'єкта  $Y = f(X)$  в багатозначному (двійковому) алфавіті з вираженим відношенням координат вхідних і вихідних змінних:

$$C = (C_1, C_2, \dots, C_i, \dots, C_n), C_i = (X_i, Y_i), C_{ij} \in \{\alpha_1, \alpha_2, \dots, \alpha_r, \dots, \alpha_k\}.$$

Двійковий алфавіт  $X_{ij} \in \{\alpha_1, \alpha_2, \dots, \alpha_r, \dots, \alpha_k\}, Y_{ij} \in \{0, 1\}$  визначення вихідних координат формує алгебру логіки, де кожне висловлення приймає значення істинності або неправди:  $f(X_i) = Y_i = \{1, 0\}$ . Тотожно істинні висловлення  $f(X_i) = Y_i = \{1\}$  формують алгебру кінцевих предикатів (АКП). Достоїнство АКП полягає у виключенні з таблиці істинності всіх рішень, що обертають предикат у помилкове висловлення, що істотно зменшує інформаційні витрати при описі поведінки дискретного об'єкта. Табличне завдання функції АКП містить всі відносини логічних змінних у явному виді. Це виключає необхідність робити обчислювальні процедури для одержання розв'язку, що характерно для рівнянь АКП, що задають відносини в неявній або аналітичній формі. Проте, існує ефективна для АКП ніша – опис процес-моделей реєстрового або векторного рівня ієрархії, які формують алгоритми обробки даних, де більш кращою є аналітична форма, представлена предикатними рівняннями. Для опису графової структури алгоритму найбільш

кращою є саме аналітична предикатна форма, як максимально наближена до візуальної, зрозумілої для людини форми, що задає послідовність дій або обчислювальних процедур для одержання розв'язку. Отже, три форми завдання предикатів мають свої ніші: 1) нижній або функціональний рівень ієрархії – поведінка об'єкта представлена предикатом-таблицею у вигляді сукупності явних рішень; 2) структурний (системний, реєстровий) рівень – процес-модель задається предикатом-рівнянням; 3) структурно-функціональний рівень – компромісний розв'язок між таблицею й рівнянням, що представлений графом взаємозалежних таблиць, де кожна з них задається в явному виді системою відносин, а графова структура – неявною або аналітичною формою у вигляді рівнянь АКП. Інтерес викликає питання ефективності подання структур даних формами: таблиця істинності, рівняння, компроміс. Таблиця має дві підмножини, що відповідають нульовій і одиничній вихідній координатам. Відповідно до закону виключення третього – якщо куб не нульовий, то він одиничний (третього не дане), розмірність таблиці істинності можна зменшити шляхом виключення одиничних або нульових рядків (кубів), що формують відповідні двійкові розв'язки по вихідних координатах. Але тоді немає необхідності зберігати одиничні або нульові координати станів виходів. У результаті таблиця істинності перетворюється в асоціацію векторів вхідних змінних, формуюче одиничне або нульове значення вихідної змінної. У цьому випадку можна оперувати поняттям матриці рівнозначних координат, які групуються в упорядковану сукупність векторів. Таким чином, будь-яка таблиця істинності  $C$  може бути перетворена у дві матриці, нульову й одиничну щодо стану вихідний змінної, які можуть відрізнятися кількістю рядків або векторів:

$$C = \{M^1 = M_{ij}^1; M^0 = M_{ij}^0\}, i = \overline{1, n_0}, r = \overline{1, n_1}, j = \overline{1, k-1}, n_0 + n_1 = n.$$

Тут одинична матриця формує структуру, що відповідає табличному завданню кінцевого предиката (DNF) (рис. 2.3):

$$P = f(X) = 1 \approx M^1 = M_{ij}^1.$$

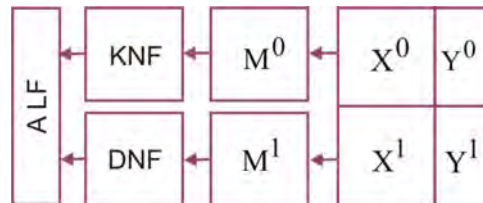


Рисунок 2.3 – Форми завдання функцій алгебри логіки

Всі рядки матриці формують тільки тотожно - щирі розв'язок, задані в явному виді. Достоїнство матричного (табличного) завдання предиката полягає у виключенні складних процедур аналізу матриці для формування розв'язок.

Обчислювальні процедури аналізу таблиць (матриць) і рівнянь істотно розрізняються по своїй складності.

1. Аналіз таблиці істинності – моделювання справного поведження. Визначення розв'язок як реакції вихідний двійкової (троїчної) змінної таблиці істинності на вхідний вектор полягає в пошуку несуперечливих результатів перетинання вектора з кожним рядком таблиці:

$$Y = Y \cup C_{ik} \leftarrow (m \cap C_i \neq \emptyset).$$

Даної моделі процесу ставиться у відповідність ізоморфна алгебрологічна структура, що формує розв'язок у двійковому алфавіті:

$$Y = Y \vee C_{ik} \leftarrow (m \wedge C_i = m).$$



З огляду на, що в таблиці істинності, заданої у двійковому алфавіті, може існувати тільки одне розв'язок, що задовольняє умові  $m \wedge C_i = m$ , функціонал обчислення стану виходу  $Y$  трансформується до більше простого виразу:

$$Y = C_{ik} \leftarrow (m \wedge C_i = m).$$

2. Аналіз одиничної матриці  $M^1$  – всі змінні рівнозначні. Формується вектор позитивної або негативної взаємодії вхідного запиту з рядками матриці. Потім визначається розв'язок  $A$  шляхом об'єднання або групування несуперечливих (позитивних) результатів перетинання вектора з кожним рядком матриці:

$$A = A \cup M_i^1 \leftarrow (m \cap M_i^1 \neq \emptyset).$$

Теоретико-множинної моделі процесу ставиться у відповідність ізоморфна алгебро-логічна структура, що формує розв'язок у двійковому алфавіті:

$$A = M_i^1 \leftarrow (m \wedge M_i^1 = m).$$

3. Більше складна процес-модель пов'язана з аналізом матриці  $M$  на основі сформованого вихідного  $m$ -вектора позитивної або негативної взаємодії запиту з рядками матриці, що представлена наступним виразом:

$$A^s = \left( \bigcap_{\forall m_i=1} M_i \setminus \bigcup_{\forall m_i=0} M_i \right); \quad A^m = \left( \bigcup_{\forall m_i=1} M_i \setminus \bigcup_{\forall m_i=0} M_i \right).$$

Тут два рівняння визначають єдині й множинні розв'язок шляхом аналізу одиничних і нульових рядків матриці, відзначених позитивним або негативним станом координат  $m$ -вектора. Ізоморфні рівняння в алгебрі логіки, що відповідають останнім теоретико-множинним розв'язокм, мають вигляд:

$$A^s = \left( \bigwedge_{\forall m_i=1} M_i \right) \wedge \left( \overline{\bigvee_{\forall m_i=0} M_i} \right); \quad A^m = \left( \bigvee_{\forall m_i=1} M_i \right) \wedge \left( \overline{\bigvee_{\forall m_i=0} M_i} \right).$$

4. Метод невизначених коефіцієнтів [60] для мінімізації функцій алгебри логіки також зводиться до аналізу матриці вхідних змінних по нульових і одиничних координатах станів вектора вихідних координат шляхом використання останніх рівнянь. Нехай є таблиця істинності функції від трьох змінних, доповненими всіма можливими сполученнями вхідних змінних для мінімізації булевої функції:

$X_1$	$X_2$	$X_3$	$X_1X_2$	$X_1X_3$	$X_2X_3$	$X_1X_2X_3$	$m \approx F$
0	0	0	00	00	00	000	1
0	0	1	00	01	01	001	0
0	1	0	01	00	10	010	1
0	1	1	01	01	11	011	0
1	0	0	10	10	00	100	1
1	0	1	10	11	01	101	0
1	1	0	11	10	10	110	0
1	1	1	11	11	11	111	1

Застосування теоретико-множинної процес-моделі

$$A^m = \left( \bigcup_{\forall m_i=1} M_i \setminus \bigcup_{\forall m_i=0} M_i \right) = \left( \bigcup_{\substack{j=1,7 \\ \forall m_i=1}} M_{ij} \setminus \bigcup_{\substack{j=1,7 \\ \forall m_i=0}} M_{ij} \right)$$

до рядків таблиці істинності дає наступний результат:

$X_1$	$X_2$	$X_3$	$X_1X_2$	$X_1X_3$	$X_2X_3$	$X_1X_2X_3$	F
.	.	.	.	00	00	000	1
.	.	.	.	00	.	010	1
.	.	.	.	.	00	100	1
.	.	.	.	.	.	111	1

Покриття одиничних рядків функції кон'юнктивними термами мінімальної вартості, що визначає оптимальну ДНФ, має такий вигляд:  $F = X_1X_2X_3 \vee \bar{X}_2\bar{X}_3 \vee \bar{X}_1\bar{X}_3$ . Квазіоптимальне покриття може бути складене з лівих по положенню в таблиці термів, що належать кожному одиничному рядку.

Н-метод мінімізації булевих функцій. Процес-модель мінімізації булевої функції по надлишковій таблиці істинності може бути отримана, якщо використовувати діаграму Хасе [59, 60] як форма, що дозволяє оптимально задавати ієрархію й еволюцію будь-якої асоціативної логічної структури. На рис. 2.4 представлені три графи, де перш і другий відповідають нульовим і одиничним рядкам таблиці істинності F, заданої вище, а третій є результатом вираховання  $G = G_i^1 \setminus G_i^0$ . Достоїнства діаграми Хасе в даному прикладі полягають в ефективному формуванні розв'язок на основі теоретико-множинного вираховання вмісту вершин нульового графа з відповідних вершин одиничного графа. Після цього мінімальне розв'язок визначається такими нижніми гранями (вершинами) третьої діаграми (див. рис. 2.4), які покривають всі рядки або куби самої верхньої грані (Supremum G).

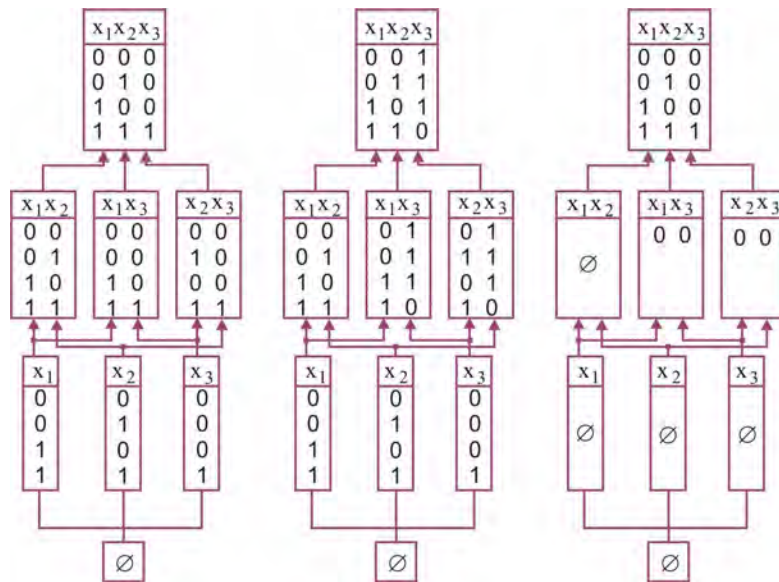


Рисунок 2.4 – Діаграми Хасе для мінімізації логічної функції

Обчислювальна складність нового Н-Методу (Hasse) мінімізації на основі використання діаграм Хасе визначається процедурами: 1) створенням двох графів  $G^1, G^0$ ; 2) вибіркою операцією покомпонентного теоретико-множинного вирахування вмісту вершин:  $G = G_i^1 \setminus G_i^0, i = \overline{1, 2^n}$ ,  $n$  – кількість змінних булевої функції; 3) пошуком квазіоптимального розв'язок шляхом покриття кубів верхньої вершини наборами, що належать нижнім компонентам графа. Формула обчислювальної складності, що включає три доданки для згаданих процедур, має вигляд:  $Q = 4n + \frac{1}{4}n^2 + \frac{1}{4}n^2 = 4n + \frac{1}{2}n^2$ . Стратегія одержання мінімальної ДНФ – за рахунок введення надмірності до таблиці істинності  $G_{nj}$  забезпечити її покриття простими кубами  $G_{ij} \in G$ , розташованими в нижніх вершинах:

$$G = G_i^1 \setminus G_i^0; F = F \cup G_{ij} \leftarrow G_{ij} \cap G_{nj} = G_{nj}.$$

Для одержання мінімальної КНФ необхідно виконати вирахування одиничного графа з нульового, а потім покрити нульові куби таблиці істинності  $G_{nj}$  більше простими кубами  $G_{ij} \in G$  компонентів, розташованих якнайнижче:

$$G = G_1^0 \setminus G_1^1; F = F \cup G_{ij} \leftarrow G_{ij} \cap G_{nj} = G_{nj}.$$

Діаграма Хасе є ідеальна, замкнута, ієрархічна, мозгоподібна структура логічних теоретико-множинних відносин у формі булеана примітивів, де вирішуються всі завдання взаємодії асоціацій змінних. З метою зменшення розмірності діаграми Хасе для її розуміння людиною необхідно виконувати розбивку числа змінних і будувати двох або ієрархічну багаторівневу структуру по можливості з однаковим числом змінних або примітивів. Наприклад, при  $n=16$  варто побудувати 4 діаграми Хасе по 4 примітиви в кожній з них, які поєднуються графом такої ж потужності, де примітивами виступають діаграми нижнього рівня ієрархії. Недолік діаграми полягає в неможливості представити операцію доповнення безпереборної процедурою. На ній визначені властивості, що дозволяють обчислювати тільки дві теоретико-множинні операції:

$$a \cap b = \inf\{a, b\}; a \cup b = \sup\{a, b\},$$

що використовують знаходження точної верхньої й нижньої грані по діаграмі [60]. Для графа Хасе, ліва частина рис 2.5, визначення доповнення має такий вигляд:  $(\bar{a} = b) \leftarrow (a \cap b = \emptyset) \& (a \cup b = Y)$ , де символ  $Y$  є універсум примітивів або універсальна множина. Це пов'язане з тим, що діаграма Хасе є структура або решітка, що має певні обмеження, пов'язані з неможливістю обчислення операції доповнення.

Структура, представлена в правій частині рис 2.5, виключає згаданий недолік. Вона являє собою вже дистрибутивна решітка з доповненнями, тому є

базисною для реалізації будь-яких обчислювальних систем. Топологічне визначення операції доповнення є перехід по діагоналі від будь-якого компонента на окружності через її центр  $\text{mid}(a)$  – middle. Бінарні операції  $(\cap, \cup)$  пов'язані з переходом по хорді окружності. Технологія аналізу С-діаграми для обчислення теоретико-множинних операцій представлена трьома виразами:

$$C(\cap, \cup, \sim) = \begin{cases} a \cap b = \inf\{a, b\}; \\ a \cup b = \sup\{a, b\}; \\ \tilde{a} = \text{mid}(a). \end{cases}$$

Це означає, що граф Хасе є часткою випадково запропонованої круговий (Circle) С-діаграми. Практична значимість введення С-Графа полягає в створенні універсального шаблону як процес-моделі для неарифметичних асоціативно-логічних завдань пошуку, розпізнавання й прийняття рішень.

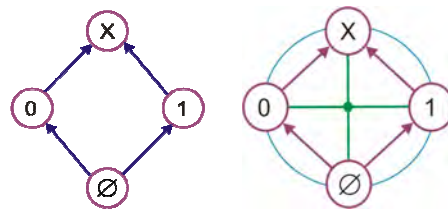


Рисунок 2.5 – Діаграми для чотиризначного алфавіту

На рис. 2.6 і 2.7 представлені діаграми Хасе й С-графи для булеанов, утворених із трьох і чотирьох примітивів відповідно. Восьмизначний алфавіт на універсумі із трьох елементів представлений наступною множиною всіх підмножин:

$$G^8 = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}.$$

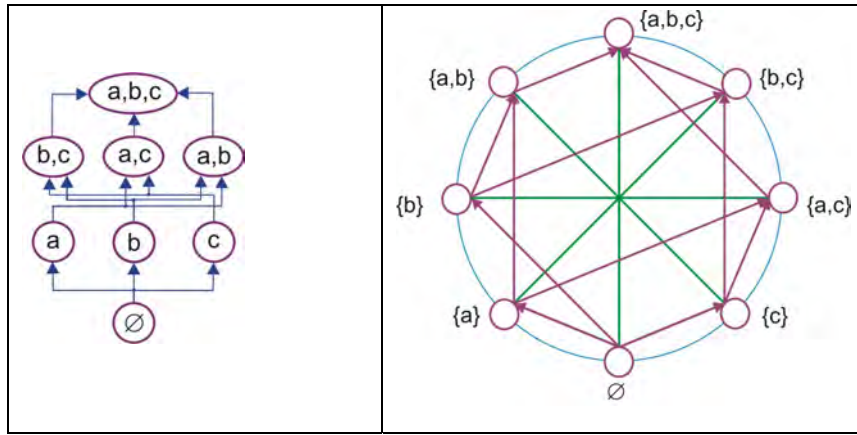


Рисунок 2.6 – Діаграми для 8-значного алфавіту

Аналітична форма 16-значного алфавіту має такий вигляд:

$$G^{16} = \{Q=00, E=01, H=10, J=11, O=\{Q, H\}, I=\{E, J\}, A=\{Q, E\}, B=\{H, J\}, S=\{Q, J\}, \\ P=\{E, H\}, C=\{E, H, J\}, F=\{Q, H, J\}, L=\{Q, E, J\}, V=\{Q, E, H\}, Y=\{Q, E, H, J\}, U\}.$$

Символ  $U$  позначає порожню множина  $\emptyset$  і служить для замикання алфавіту щодо теоретико-множинних операцій і його зв'язку з математичними структурами більше високої ієрархії, а також для побудови систем доведення.

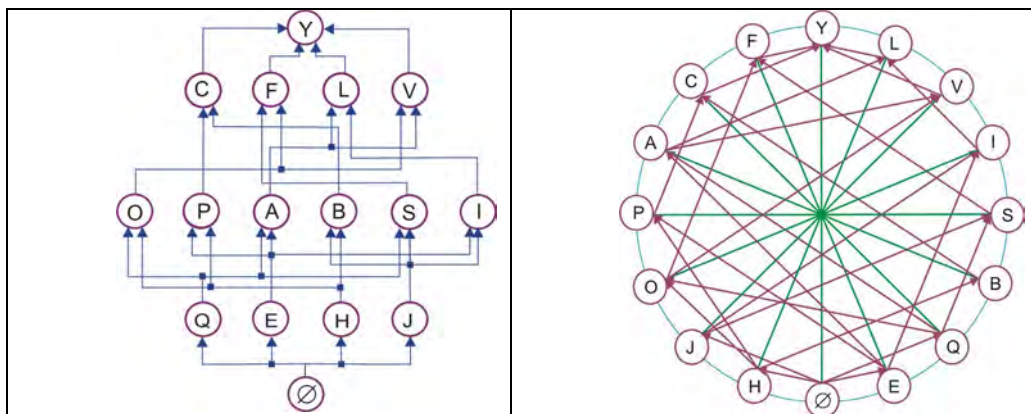


Рисунок 2.7 – Діаграми для 16-значного алфавіту

Алфавіт  $G^{16}$  задає всі можливі переходи (відношення) на універсумі примітивів  $Y = \{00, 01, 10, 11\}$ . Визначення замкнутого щодо теоретико-множинних операцій (перетинання, об'єднання, доповнення) алфавіту пов'язане з генеруванням булеана (множинаі всіх підмножин) на універсумі  $Y$ . Потужність породжуваного в такий спосіб алфавіту визначається вираженням  $\eta = 2^n$ , де  $n$  – кількість примітивів у сімействі  $Y$ . Запропонований алфавіт створює універсальний математичний апарат для синтезу й аналізу функцій і структур, що використовують бінарну концепцію поводження об'єкта.

Недолік С-діаграми полягає у відсутності чітко вираженої ієрархії символів, що є присутнім у діаграмі Хасе. Можна використовувати ідею центральної симетрії для діаграми Хасе. Вона має взаємно доповнюють, до потужності універсума примітивів  $n$ , рівні ієрархії (див. рис 2.7):  $0+4=4$ ,  $1+3=4$ ,  $2+2=4$ . Число рівнів дорівнює  $n+1$ , включаючи рівень порожньої множині. Таким чином, для діаграми Хасе, представленої на рис. 2.7, вертикальна симетрія щодо горизонтального рівня 2 дає можливість визначити операцію доповнення у вигляді пар символів: U-Y, Q-C, E-F, H-L, J-V, A-B, P-S, O-I. Останні три пари мають горизонтальну симетрію усередині середнього другого рівня щодо вертикалі, що розділяє його на два рівних по потужності підмножини, що містять у цьому випадку по трьох елемента. Отже, завжди можна побудувати таку топологію діаграми Хасе, що буде містити операцію доповнення у вигляді симетричних символів, що взаємно доповнюють один одного до універсума примітивів.

### 2.3 Метрика алгебри векторної логіки

Векторний дискретний логічний (булев) простір визначає взаємодія об'єктів шляхом використання трьох аксіом (тотожності, симетрії й трикутника), що формують неарифметичну В-метрику векторного виміру [99]:



$$B = \begin{cases} d(a, b) = a \oplus b = (a_i \oplus b_i), i = \overline{1, n}; \\ d(a, b) = [0 \leftarrow \forall i (d_i = 0)] \leftrightarrow a = b; \\ d(a, b) = d(b, a); \\ d(a, b) \oplus d(b, c) = d(a, c), \\ \oplus = [d(a, b) \wedge \bar{d}(b, c)] \vee [\bar{d}(a, b) \wedge d(b, c)]. \end{cases}$$

Вершини в транзитивному трикутнику  $(a, b, c)$  (рис. 2.8) є вектори, що ідентифікують об'єкти в  $n$ -мірному булевому  $B$ -просторі, сторони трикутника  $d(a, b)$ ,  $d(b, c)$ ,  $d(a, c)$  є відстані між вершинами, які також представлені векторами розмірності  $n$ , де кожний розряд визначений у тій же алфавіті, що й координати векторів-вершин.

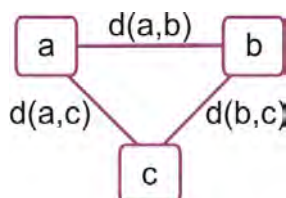


Рисунок 2.8 – Трикутник векторного транзитивного замикання

Векторний транзитивний трикутник має повну аналогію чисельному виміру відстані в метричному  $M$ -просторі, що задається системою аксіом, що визначає взаємодію однієї, двох і трьох крапок у будь-якому просторі:

$$M = \begin{cases} d(a, b) = 0 \leftrightarrow a = b; \\ d(a, b) = d(b, a); \\ d(a, b) + d(b, c) \geq d(a, c). \end{cases}$$

Специфіка аксіоми метричного трикутника полягає в чисельному (скалярному) порівнянні відстаней трьох об'єктів, коли інтервальна невизначеність відповіді – дві сторони трикутника можуть бути більше або рівні третій – малопригодна для визначення точної довжини останньої

сторони. Усунути даний недолік можна тільки в логічному векторному просторі, що може мати детерміноване подання про кожний параметр стану процесу або явища. Тоді чисельна невизначеність третьої сторони трикутника у векторному логічному просторі здобуває форму точного двійкового вектора, що характеризує відстань між двома об'єктами й обчислюється на основі знання відстаней двох інших сторін трикутника:

$$d(a, b) \oplus d(b, c) = d(a, c).$$

Три аксіоми визначення метрики надлишкові, принаймні для векторного простору, де можна використовувати тільки одну єдину – взаємодію трьох крапок:  $d(a, b) \oplus d(b, c) \oplus d(a, c) = 0$ . З даного закону випливають дві тотожності, що визначають відносини для однієї й двох точок простору:

$$d(a, b) \oplus d(b, c) \oplus d(a, c) = 0 \rightarrow \begin{cases} d(a, b) = d(b, a) = 0 \rightarrow c = \emptyset; \\ d(a, a) = 0 \rightarrow \{b, c\} = \emptyset. \end{cases}$$

Тут також цікавим представляється наступний факт. З огляду на циклічність трикутника, по будь-яким двох відомим суміжним (інцидентним) компонентах можна обчислювати третій. Це ставиться як до станів або код вершин, так і до відстаней між ними:

$\begin{cases} d(a, b) = d(a, c) \oplus d(b, c) \\ d(b, c) = d(a, b) \oplus d(a, c) \\ d(a, c) = d(a, b) \oplus d(b, c) \end{cases}$	$\begin{cases} d(b, c) = b \oplus c \\ d(a, c) = a \oplus c \\ d(a, b) = a \oplus b \end{cases}$	$\begin{cases} a = d(a, b) \oplus b \\ b = d(b, c) \oplus c \\ c = d(c, a) \oplus a \end{cases}$
--	--	--

Ізоморфізм теорії множин стосовно алгебри логіки дозволяє так само визначити векторний теоретико-множинний S-простір, де аксіома трикутника

задається симетричною різницею  $\Delta$ , що є аналогом операції хог у булевій алгебрі:

$$S = \begin{cases} d(a, b) = a\Delta b = (a_i \Delta b_i), i = \overline{1, n}; \\ d(a, b) = [\emptyset \leftarrow \forall i(d_i = \emptyset)] \leftrightarrow a = b; \\ d(a, b) = d(b, a); \\ d(a, b)\Delta d(b, c) = d(a, c), \\ \Delta = [d(a, b) \cap \tilde{d}(b, c)] \cup [\tilde{d}(a, b) \cap d(b, c)]. \end{cases}$$

Тут  $\Delta$ -операція симетричної різниці на чотиризначному теоретико-множинному алфавіті Кантора  $\alpha = \{0, 1, x = \{0, 1\}, \emptyset\}$  представлена наступною таблицею:

$\Delta$	0	1	x	$\emptyset$
0	$\emptyset$	x	1	0
1	x	$\emptyset$	0	1
x	1	0	$\emptyset$	x
$\emptyset$	0	1	x	$\emptyset$

При визначенні відстані між двома векторами в S-Просторі використовується симетрична різниця, що ізоморфна хог-операції в булевому B-просторі. Приклади обчислення відстаней між векторами в обох просторах (S, B) наведені нижче:

$$S = \begin{array}{c|cccccccc} & a & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ & b & x & x & 0 & 0 & 1 & 1 & 0 & 0 \\ & c & x & x & x & x & 0 & 0 & 0 & 0 \\ \hline d(a, b) & 0 & 1 & \emptyset & \emptyset & \emptyset & x & \emptyset & x \\ d(b, c) & \emptyset & \emptyset & 1 & 1 & x & x & \emptyset & \emptyset \\ d(a, c) & 0 & 1 & 1 & 1 & x & \emptyset & \emptyset & x \end{array}$$

$$B = \begin{array}{c|cccccccc} & a & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ & b & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ & c & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline d(a, b) & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ d(b, c) & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ d(a, c) & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}$$

Вектор, дорівнює нулю (порожній множині) по всіх координатах, означає повний збіг результату із запитом, так само як і вектор, дорівнює

одиниці (символу  $x$ ) по всіх розрядах, свідчить про повну суперечливість результату запиту.

Кількість градацій один змінної може бути кінцевим і кратним ступеня 2 числом  $\alpha = 2^n \rightarrow \{2^2 = 4, 2^4 = 16\}$ , що визначається потужністю булеана на універсумі з  $n$  примітивів. Інакше, симетрична різниця може існувати тільки на замкнутому щодо теоретико-множинних операцій алфавіті. Таким чином, взаємодія двох об'єктів у векторному логічному просторі може мати як двійкову, так і багатозначну детерміновану шкалу виміру взаємодії. Діаграма Хасе від будь-якого кінцевого числа примітивів (1,2,3,4, ...) може бути захована в змінну логічного вектора. Більше того, 16 градацій, наприклад, взаємодії векторів на чотирьох примітивах точно показують не тільки ступінь близькості по розглянутій змінній, але й у чому саме вони розходяться - по яких примітивах або їхньому сполученні.

З метою перевірки помилок передачі даних корисної може бути процедура згортки відстаней для числа векторів, рівних ступені 2:1) Обчислити всі відстані між двійковими кодами, включаючи останній і перший вектори, у результаті чого виходить замкнута геометрична фігура  $c_i = a_i \oplus a_{i+1} (i = n \rightarrow i + 1 = 0)$ . 2) Обчислити всі відстані між непересічними парами отриманих на першому етапі кодів  $c_i = a_{2i-1} \oplus a_{2i} (i = 1, 2, 3, \dots, n)$ . 3) Повторювати процедуру 2 до одержання одного коду, рівного нулю по всіх координатах. Аналогічні дії можна виконати й для багатозначних векторів, де, наприклад, кожна координата визначена в чотиризначному теоретико-множинному алфавіті, а процедура зводиться до одержання вектора порожніх значень координат. Тут відбувається згортка замкнутого простору до однієї крапки (рис 2.9), певної по всіх координатах символами порожньої множині, шляхом обчислення відстані між вектор-об'єктами, а потім - між векторами-відстанями. Інакше, сума по модулі всіх векторів-відстаней, замкнутих у цикл, дорівнює порожньому вектору

$$m_i = c_i \oplus_{j=|i+1|_n}^{i-1} c_j \rightarrow m = m_i \oplus_{i=1}^{n-1} m_{i+1}.$$

Але така процедура має меншу глибину діагностування помилок - можлива фіксація невірного розряду, у той час як по двійковому дереву згортання простору є можливість підвищити глибину діагностування до пари векторів.

Згортка простору становить інтерес для багатьох практичних завдань: 1) Діагностування й виправлення помилок при передачі інформації з каналів зв'язку. 2) Пошук дефектів у цифрових виробках на основі таблиць (функцій) несправностей. 3) Пошук дефектів у цифрових виробках на основі багатозначних таблиць (функцій) несправностей.

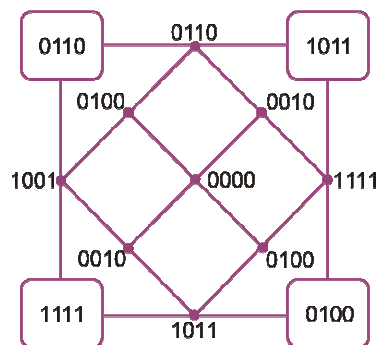


Рисунок 2.9 – Згортка замкнутого простору

Сутність згортки простору закладено в метриці транзитивного трикутника, який можна перетворити шляхом переносу правої частини рівності в ліву:  $d(a, b) \oplus d(b, c) = d(a, c) \rightarrow d(a, b) \oplus d(b, c) \oplus d(a, c) = 0$ .

Далі представлене теоретичне обґрунтування згортання простору шляхом доведення коректності використання метрики векторного булевого простору з метою виміру взаємодії логічних конструкцій, включаючи точку, лінію, площину.

**Аксиома 2.1.** Для булевих змінних має місце логічний вираз:  $a = b \rightarrow a \oplus b = 0$ . Справді, сума за модулем 2 є функція нерівнозначності, що приймає щире або одиничне значення при розбіжності значень змінних  $a \neq b \rightarrow a \oplus b = 1$  і нульове – при рівності аргументів.

**Визначення 2.1.** Відстань між двома точками  $n$ -мірного простору є вектор, обчислена на основі додавання за модулем 2 (далі хор-сума) однойменних координат:

$$d(a, b) = a_i \oplus_{i=1}^n b_i.$$

**Визначення 2.2.** Вектор-відстань між двома точками (об'єктами)  $n$ -мірного логічного простору дорівнює нулю (одиниці), якщо всі координати вектора дорівнюють нулю (одиниці):

$$d(a, b) = 0(1) \leftarrow \forall i [a_i \oplus_{i=1}^n b_i = 0(1)].$$

**Визначення 2.3.** Простий ланцюг – послідовність векторних відстаней і крапок простору, серед яких немає повторюваних компонентів: відстаней, крапок.

**Визначення 2.4.** Векторно-логічний цикл  $D$  є сукупність векторних відстаней  $d_i \in D$  між точками простору, що створюють замкнутий простий ланцюг, у якій перша й остання точки простору збігаються.

**Теорема 2.1.** Хор-сума векторних відстаней, що утворюють цикл, між двома точками  $n$ -мірного простору завжди дорівнює нулю:  $d(a, b) \oplus d(b, a) = 0$ .

Це слідує з аксіоми симетрії (комутативності), яка припускає, що вектор-відстані між двома будь-якими точками  $n$ -мірного простору  $d(a, b) = d(b, a)$  рівні між собою. Але, відповідно до аксіоми 2.1, перенос правої частини

рівності в ліву супроводжується регулюванням відношень між компонентами за допомогою xor-операції, результат якої дорівнює нулю:  $d(a, b) \oplus d(b, a) = 0$  через рівність векторних відстаней.

**Теорема 2.2.** Xor-сума векторних відстаней або двох сторін у транзитивному трикутнику завжди дорівнює третій.

Доведення. У загальному випадку метрика визначає взаємодію трьох точок  $\{a, b, c\}$  (сторін трикутника) у просторі шляхом формування трьох відстаней  $d(a, b), d(b, c), d(a, c)$ . Допустимо, що є дві відстані  $d(a, b), d(b, c)$ , які, відповідно до теореми 2.1, створюють рівність  $d(a, b) \oplus d(b, c) = 0$ , що має три варіанти взаємодії точок:

1)  $d(a, b) = d(b, c) = 0$  – є одна точка, позначена в цьому випадку ідентифікаторами  $a, b, c$ , відстань між якими дорівнює нулю;

2)  $d(a, b) = d(b, c) \rightarrow d(a, b) \oplus d(b, c) = 0$  – є дві точки  $\{b, a=c\}$ , які формують дві однакових відстані, що створюють цикл, відповідно до принципу симетрії або комутативності;

3)  $d(a, b) \neq d(b, c) \rightarrow d(a, b) \oplus d(b, c) \neq 0 \rightarrow d(a, b) \oplus d(b, c) = d(a, c)$  – є два нерівних між собою відстані  $d(a, b) \neq d(b, c)$ , які можливі лише при взаємодії трьох точок простору  $a, b, c$  з метою визначення третьої відстані за допомогою векторної операції  $d(a, b) \oplus d(b, c) = d(a, c)$ . При цьому вектор, заданий у правій частині рівності, ніколи не буде дорівнювати одному зі складу лівої частини через те, що  $d(a, b) \neq d(b, c)$ . Таким чином, відносини трьох будь-яких крапок векторного логічного простору можуть бути зведені до формальної взаємодії, записаній за допомогою рівності  $d(a, b) \oplus d(b, c) = d(a, c)$ , що, вироджуючись, також регулює співвідношення двох і однієї крапки, самої на себе.

**Теорема 2.3.** Xor-сума векторних відстаней у транзитивному трикутнику дорівнює нулю:  $d(a, b) \oplus d(b, c) = d(a, c) \rightarrow d(a, b) \oplus d(b, c) \oplus d(a, c) = 0$ . Доведення ґрунтується на застосуванні аксіоми до виразу транзитивного замикання, отриманому в теоремі 2.2.

**Теорема 2.4.** Хор-сума  $\bigoplus_{i=1}^n d_i = 0$  векторних відстаней  $d_i \in D$  у циклі  $D$ ,

складеному з кінцевого числа вершин ( $n$ ), дорівнює нулю:

- 1)  $d_1 = 0$ ;
- 2)  $d_1 \oplus d_2 = 0$ ;
- 3)  $d_1 \oplus d_2 \oplus d_3 = 0$ ;
- 4)  $d_1 \oplus d_2 \oplus d_3 \oplus d_4 = 0$ ;
- 5)  $d_1 \oplus d_2 \oplus \dots \oplus d_i \oplus \dots \oplus d_n = 0$ .

Доведення слідує із застосування теорем 2.1-2.3 до відстаней між точками (вершинами), складовими замкнуті цикли. У першому випадку фіксується відстань транзитивного замикання точки самої на себе. У другому – відстань між двома транзитивно замкнутими точками. У третьому – між трьома точками простору. У четвертому – між чотирма точками. П'ятий випадок узагальнює наявність нульової відстані в сумі транзитивних замикань будь-якого числа точок в  $n$ -мірному векторному просторі.

**Слідство 2.1.** Хор-Сума будь-яких двійкових кодів однакової довжини дорівнює нулю, якщо вони становлять цикл.

**Слідство 2.2.** Метрика  $\beta$  векторного логічного простору визначається єдиною рівністю, що формує нульову хор-суму відстаней між ненульовим і кінцевим числом крапок, замкнутих у цикл:

$$\beta = \bigoplus_{i=1}^n d_i = 0.$$

**Визначення 2.5.** Метрика  $\beta$  векторного логічного простору є рівна нулю хор-сума відстаней між кінцевим числом точок циклу.

Дане визначення ставить в основу не елементи множини, але відношення, що дозволяє скоротити систему аксіом метрики із трьох до однієї й



поширити її дію на як завгодно складні конструкції  $n$ -мірного простору. Класичне завдання метрики для визначення взаємодії однієї, двох і трьох точок у векторному логічному просторі, є часткою випадку  $\beta$  метрики при  $i = 1, 2, 3$  відповідно:

$$M = \begin{cases} d_1 = 0 \leftrightarrow a = b; \\ d_1 \oplus d_2 = 0 \leftrightarrow d(a, b) = d(b, a); \\ d_1 \oplus d_2 \oplus d_3 = 0 \leftrightarrow d(a, b) \oplus d(b, c) = d(a, c). \end{cases}$$

Зокрема, метричний, функціональний і інший види просторів у сумі також дають нуль. Наприклад, фігура зі сторонами 1, 2, 3, згідно всім шкільним підручникам, не є трикутник, оскільки три точки розташовані на одній прямій (рис 2.10). Але аксіома транзитивного замикання метрики використовує структуру, складену із трьох точок на площині з різними координатами, що строго зветься трикутника. Тоді фігура зі сторонами 1, 2, 3, відповідно до визначення метрики, є в чистому виді трикутник із двома нульовими кутами й третім, рівним 180 градусів, де повністю виконуються умови для трьох сторін:  $a + b \geq c \rightarrow 1 + 2 = 3$ .

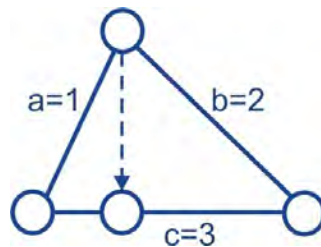


Рисунок 2.10 – Метричний трикутник

## 2.4 Аналіз табличних структур даних

Розглядається таблиця  $A$ , розмірністю  $m \times n$ , взаємодіюча із двома векторами  $m_a, m_b$ , які можуть бути вхідними й вихідними, залежно від процедури аналізу (рис 2.11):

$$A = (A_1, A_2, \dots, A_i, \dots, A_n); \quad A = (A_1, A_2, \dots, A_j, \dots, A_m);$$

$$m_a = (m_1, m_2, \dots, m_r, \dots, m_n); \quad m_b = (m_1, m_2, \dots, m_s, \dots, m_m).$$

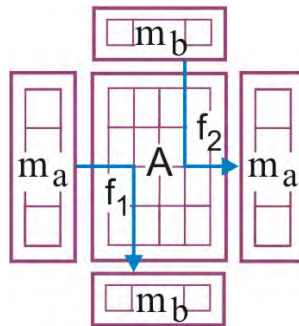


Рисунок 2.11 – Первинний аналіз таблиці

Первинний аналіз: запит – таблиця – відповідь:

$$m_a \rightarrow A \rightarrow m_b \vee m_b \rightarrow A \rightarrow m_a$$

пов'язаний з виконанням процес-моделей  $f_1 \vee f_2$ , де результатом є множина рішень, відзначених нульовими координатами вихідного двійкового вектора, якщо відповідний стовпець (рядок) таблиці задовольняє вхідному запиту, і одиничними – у протилежному випадку:

$$m_b = f_1(m_a, A) \rightarrow m_{bj} = m_a \bigoplus_{j=1}^m A_j; \quad m_a = f_2(m_b, A) \rightarrow m_{ai} = m_a \bigoplus_{i=1}^n A_i.$$

Тут операція хог може бути замінена симетричною різницею для аналізу багатозначних таблиць у замкнутому теоретико-множинному алфавіті:

$$m_b = f_1(m_a, A) \rightarrow m_{bj} = m_a \Delta_{j=1}^m A_j; \quad m_a = f_2(m_b, A) \rightarrow m_{ai} = m_a \Delta_{i=1}^n A_i.$$

При цьому якість взаємодії векторів буде мати багатозначну градацію, що зменшується в міру зростання повноти символу замкнутого алфавіту. Наприклад, для чотиризначного алфавіту Кантора  $\langle \emptyset \rightarrow \{0,1\} \rightarrow x \rangle$  або 16-значного алфавіту опису автоматних змінних  $\langle \emptyset \rightarrow \{Q, E, H, J\} \rightarrow \{O, I, A, B, S, P\} \rightarrow \{C, F, L, V\} \rightarrow Y \rangle$ .

Вторинний аналіз таблиці зв'язаний зі згорткою стовпців або рядків таблиці для одержання інтегрованого розв'язок у формі одного вектора, що означає підвищення глибини пошуку інформації за рахунок виключення з узагальненого вектора суперечливих (неможливих) станів координат шляхом виконання процедур:

$$m_a = f_1^s(m_b, A) \rightarrow \bigwedge_{\forall j(m_{bj}=0)} A_j \wedge \overline{\bigvee_{\forall j(m_{bj}=1)} A_j}; \quad m_a = f_1^m(m_b, A) \rightarrow \bigvee_{\forall j(m_{bj}=0)} A_j \wedge \overline{\bigvee_{\forall j(m_{bj}=1)} A_j};$$

$$m_b = f_2^s(m_a, A) \rightarrow \bigwedge_{\forall i(m_{ai}=0)} A_i \wedge \overline{\bigvee_{\forall i(m_{ai}=1)} A_i}; \quad m_b = f_2^m(m_a, A) \rightarrow \bigvee_{\forall i(m_{ai}=0)} A_i \wedge \overline{\bigvee_{\forall i(m_{ai}=1)} A_i}.$$

Споконвічно вектори в таблиці формують загальний простір, відзначений нульовими й одиничними значеннями вектора-запиту. Останні процедури дозволяють розділити одиничний і нульовий простори друг від друга й

сформувати множина тільки істотних координат в одному нульовому або одиничному векторі. Відмінність функціоналів  $f^s, f^m$  полягає в пошуку одиничних координатних рішень шляхом перетинання всіх рядків у першому випадку, або у визначенні множинного розв'язок на основі об'єднання всіх рядків таблиці – у другому випадку. Аналогічно можна відокремити від нульового простору одиничне й сформувати одиничні відповідні вектори істотних координат:

$$m_a = f_1^s(m_b, A) \rightarrow \bigwedge_{\forall j(m_{bj}=1)} A_j \wedge \overline{\bigvee_{\forall j(m_{bj}=0)} A_j}; \quad m_a = f_1^m(m_b, A) \rightarrow \bigvee_{\forall j(m_{bj}=1)} A_j \wedge \overline{\bigvee_{\forall j(m_{bj}=0)} A_j};$$

$$m_b = f_2^s(m_a, A) \rightarrow \bigwedge_{\forall i(m_{ai}=1)} A_i \wedge \overline{\bigvee_{\forall i(m_{ai}=0)} A_i}; \quad m_b = f_2^m(m_a, A) \rightarrow \bigvee_{\forall i(m_{ai}=1)} A_i \wedge \overline{\bigvee_{\forall i(m_{ai}=0)} A_i}.$$

Для багатозначної таблиці  $A$  вторинний аналіз також виконується шляхом використання останніх восьми логічних виразів, які вже оперують багатозначною логікою. Для чотиризначного алфавіту Кантора примітиви *and*, *or*, *not* визначені наступними таблицями:

$\wedge$	0	1	x	$\emptyset$	$\vee$	0	1	x	$\emptyset$
0	0	$\emptyset$	0	$\emptyset$	0	0	x	x	0
1	$\emptyset$	1	1	$\emptyset$	1	x	1	x	1
x	0	1	x	$\emptyset$	x	x	x	x	x
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	0	1	x	$\emptyset$

a	0	1	x	$\emptyset$
$\bar{a}$	1	0	$\emptyset$	x

## 2.5 Спрощення критерію якості

Для оцінки взаємодії між двома векторами  $(a, b)$  був уведений інтегральний критерій якості у векторному булевому просторі, що дозволяє визначати: близькість за Хемінгом двох об'єктів в  $n$ -мірному просторі як

потужність множині порожніх перетинань, а також векторно-логічну міру приналежності об'єктів один одному:

$$Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m),$$

$$d(m, A) = m \oplus A; \mu(m \in A) = A \wedge \overline{m} \wedge A; \mu(A \in m) = m \wedge \overline{m} \wedge A.$$

Даний критерій якості однозначно визначає три форми взаємодії двох будь-яких об'єктів в n-мірному векторному логічному просторі: відстань і дві приналежності. Однак з огляду на, що всі три оцінки входні в інтегральний критерій, об'єднані по функції  $or$ , бажання спростити інтегральну форму взаємодії векторів приводить оцінку до цікавого й компактного виду:

$$\begin{aligned} Q &= d(m, A) \vee \mu(m \in A) \vee \mu(A \in m) = m \oplus A \vee A \wedge \overline{m} \wedge A \vee m \wedge \overline{m} \wedge A = \\ &= m \oplus A \vee [A \wedge (\overline{m} \vee \overline{A})] \vee [m \wedge (\overline{m} \vee \overline{A})] = m \oplus A \vee [A\overline{m} \vee A\overline{A} \vee m\overline{m} \vee m\overline{A}] = \\ &= (A\overline{m} \vee m\overline{A}) \vee [A\overline{m} \vee A\overline{A} \vee m\overline{m} \vee m\overline{A}] = A\overline{m} \vee m\overline{A} \vee A\overline{m} \vee A\overline{A} \vee m\overline{m} \vee m\overline{A} = \\ &= m \oplus A. \end{aligned}$$

Таким чином, критерій якості  $Q = m \oplus A$  тепер погодиться з метрикою оцінювання відстані або взаємодії у векторно-логічному просторі, а також має тривіальну обчислювальну форму для оцінювання численних рішень, пов'язаних з аналізом і синтезом інформації. Кодова відстань за Хемінгом для булевого простору – занадто груба оцінка! Однаково, що вимірювати відстань у літрах бензину, що спалюється автомобілем. Інструментарій для виміру взаємодії об'єктів у просторі повинен оперувати такою ж метрикою або одиницею виміру, у який представлені самі об'єкти, і навпаки. Одиниця виміру векторного логічного простору – двійковий вектор, теоретико-множинного – багатозначний вектор. В останньому випадку алфавіт повинен бути замкнутим щодо теоретико-множинних операцій (об'єднання, перетинання, доповнення) і являти собою булеан на будь-якому універсумі примітивів.

Транзитивність трикутника надає прості правило для кодування вершин змістовних графів керування операційними автоматами. Логічний векторний простір не повинне мати метричної відстані й критеріїв якості, що включають арифметичні операції на скалярних величинах. Далі представлені практично орієнтовані постулати, що пояснюють деякі переваги векторно-логічної метрики:

Метрика, традиційна, здатна визначати відстань між геометричними центрами однорідних об'єктів. Вона не здатна визначати ступінь близькості (взаємодії) об'єктів різного рівня ієрархії (точка в колі – саме коло). Чи існує в геометрії відстань між точкою й більше складною фігурою, який належить ця крапка, лінія, площина? Відстань можна вимірювати між математичними крапками або між центрами однорідних фігур, що знову приводиться до двох математичних крапок.

Векторно-логічний критерій якості (ВЛК) відносин об'єктів різних рівнів ієрархії (точки, лінії, площини, просторові фігури) є метрика векторного логічного простору, що вимірює взаємодію векторів однієї розмірності шляхом обчислення логічного вектора  $Q = m \oplus A$ .

ВЛК визначає не тільки відстань між об'єктами векторно-логічного простору, але й взаємну приналежність об'єктів один одному:  $d(a, b) \vee \mu(a, b) \vee \mu(b, a)$ , якщо така є.

Відстань – це окремий випадок непересічної взаємодії однорідних (математичних) об'єктів. Для пересічних об'єктів відстань між ними, рівне нулю, не визначає взаємодію, що ідентифікується критерієм якості.

Взаємодія об'єктів різного рівня ієрархії при нульовій відстані визначається ступенем невизначеності присутності примітива в об'єкті, що включає даний елемент. Ступінь невизначеності присутності дефекту в цифровій системі на кристалі, що містить 700 000 000 вентилів, набагато вище, ніж у вентилі, що містить два входи й один вихід.

Знання однієї функції приналежності  $\mu(a,b)$  абсолютно не доставляє інформацію про те, що являє собою об'єкт  $b$ . Можливо, він дорівнює  $a$ . Можливо, він у мільйон разів більше елемента  $a$ . Введення додаткової функції приналежності  $\mu(b,a)$  разом з іншими складовими дає повну картину взаємодії довільних (гетерогенних) об'єктів по природі й ієрархії.

Міра невизначеності присутності примітива в об'єкті більше високого рівня ієрархії є умову для пошуку елемента шляхом розподілу даного простору з метою більше точної локалізації елемента в об'єкті. Тут повна воля для формулювання розв'язок завдань технічної діагностики у векторно-логічному просторі, без використання арифметики.

Діаграма Хасе є причинно-наслідкова інфраструктура асерційно-темпоральної логіки для генерування паралельно-послідовних маршрутів розв'язок логічних мозгоподібних задач, заснованих на базових методах-примітивах. Такими можуть бути: пошук, розпізнавання, оцінювання. Ухвалення розв'язок ґрунтується на вже підготовленій інформації, орієнтованій на бінаризацію дії. За аналогією з EDA (Design for testability, for manufacturability) примітивні, початкові процеси (пошук, розпізнавання) повинні бути орієнтовані на можливість ухвалення правильного розв'язок за мінімальний час.

## 2.6 Висновки

1. Векторний дискретний логічний (булев) простір неарифметичної В-метрики векторного виміру, що регулює взаємодію об'єктів за допомогою аксіом тотожності, симетрії й трикутника. Запропонований векторно-логічний критерій якості взаємодії векторів, що дозволяє точно й адекватно оцінювати розв'язок в процесі пошуку й розпізнавання будь-яких об'єктів у векторному логічному просторі.

2. Структури даних і процес-моделі, що використовують властивості розробленої алгебраїчної структури для асоціативного логічного аналізу, мінімізації булевих функцій, прийняття рішень без використання арифметичних операцій з метою істотного підвищення швидкодії.

3. Модифікована діаграма Хасе, що відрізняється введенням операції доповнення, необхідної для виконання повного аналізу асоціативних логічних структур даних, використовуваних у завданнях пошуку, розпізнавання й прийняття рішень.

4. Практична значимість запропонованих результатів досліджень полягає в орієнтації уведеного векторного логічного простору, процес-моделей, інфраструктури асоціативно-логічного аналізу на ефективний, точний і адекватний пошук, розпізнавання й діагностування на основі використання мультипроцесорної неарифметичної комп'ютерної системи, що дозволяє істотно (x100) підвищити швидкодія векторних логічних обчислювальних процедур.



### **3 МОДЕЛІ ГЕНЕРАЦІЇ ТЕСТІВ І МЕТОДИ ДІАГНОСТУВАННЯ SOC-КОМПОНЕНТІВ**

Пропонується структурна модель відношень компонентів (функціональність, пристрій, тест, дефекти), що дозволяє визначити й класифікувати шляхи розв'язок практичних задач, включаючи синтез тестів, моделювання несправностей і пошук дефектів [106, 115, 119, 121]. Описано методи синтезу тестів для функціональностей, заданих матричними формами опису поведінки цифрових компонентів, які відрізняються паралелізмом векторних операцій над таблицями [103, 104, 106, 109, 115, 117-120]. Запропоновано вдосконалені методи пошуку функціональних порушень, які дають можливість істотно підвищити швидкодія обчислювальних процедур, пов'язаних з діагностуванням і відновленням працездатності програмних і апаратних продуктів [102, 105, 110-114]. Моделі й методи синтезу тестів для функціональностей і діагностування порушень можуть бути використані як убудовані компоненти інфраструктури сервісного обслуговування цифрових систем на кристалах із застосуванням стандарту граничного сканування IEEE 1500.

Актуальність. Тенденція останніх років у частині створення нових комунікаційних, обчислювальних і інформаційних сервісів, корисних для людини, звертає увагу на створення усе більше спеціалізованих гаджетів (gadget), що мають істотні переваги перед персональними комп'ютерами й ноутбуками: енергоспоживання, компактність, вага, вартість, функціональні можливості, дружність інтерфейсу. Практично вся десятка кращих за 2010 рік спеціалізованих виробів (Apple iPad, Samsung Galaxy S, Apple MacBook Air, LogitechRevue, GoogleNexusOne(HTC Desire), Apple iPhone 4, Apple TV, Toshiba Libretto W100, Microsoft Kinect, Nook Color) реалізована у вигляді цифрових систем на кристалах. Світовий ринок таких гаджетів нараховує сьогодні порядку 3 мільярдів виробів. Для їхнього ефективного проектування,

виробництва й експлуатації створюються нові технології й інфраструктури сервісного обслуговування. Один з можливих кроків у даному напрямку описаний нижче.

Мета – розробка нових технологій сервісного обслуговування компонентів цифрових систем на кристалах, орієнтованих на розв’язок задач синтезу тестів і пошуку дефектів у програмних і апаратних продуктах.

Задачі: 1) Створення структурної моделі відношень компонентів (функціональність, пристрій, тест, дефекти) для визначення й класифікації шляхів розв’язку практичних задач, включаючи синтез тестів, моделювання несправностей і пошук дефектів. 2) Розробка методів синтезу тестів для функціональностей, заданих матричними формами опису поведінки цифрових компонентів. 3) Удосконалення методу пошуку функціональних порушень для підвищення швидкодії обчислювальних процедур діагностування програмних і апаратних продуктів.

Джерела: 1. Модель формулювання завдань технічної діагностики [70-74]. 2. Синтез тестів для функціональностей [68, 73-76]. 3. Убудоване діагностування цифрових систем [77-83]. 3. Апаратура й матричні процесори [84-87].

### 3.1. Моделі генерації тестів для функціональних елементів

Пропонуються технологічні й ефективні процес-моделі синтезу тестів і їхнє застосування для діагностування функціональних порушень у програмні й/або апаратних продуктах. Використовуються табличні структури даних, які орієнтовані на паралельне виконання векторних логічних операцій при генерації тестів і пошуку дефектних компонентів [117, 119].

Проблема синтезу або аналізу довільної системи може бути сформульована у вигляді взаємодії (симетричної різниці) у кібернетичному просторі її моделі  $F$  із вхідними впливами  $T$  і реакціями  $L$ :

$$f(F, T, L) = \emptyset \rightarrow F\Delta T\Delta L = \emptyset.$$

Кібернетичний простір (структура) є сукупність взаємозалежних компонентів, які із заданою адекватністю описують процеси або явища для досягнення поставлених цілей. Кожний компонент представлений  $n$ -мірною фігурою, де координата визначена в багатозначному алфавіті, наприклад Кантора  $A = \{0, 1, x, \emptyset\}$ ,  $x = \{0, 1\}$ . Мірність формує структуру фігури:  $n=1$  – вектор довжини  $k$ ;  $n=2$  – матриця розмірністю  $k_1 \times k_2$ ;  $n=3$  – куб розмірністю  $k_1 \times k_2 \times k_3$ .

Метрика  $\beta$  кібернетичного або векторного логічного двійкового простору визначається єдиною рівністю, що формує нульову хог-суму відстаней між ненульовим і кінцевим числом точок, замкнених у цикл:

$$\beta = \bigoplus_{i=1}^n d_i = 0.$$

Інакше: метрика  $\beta$  векторного логічного двійкового простору є рівна нулю хог-сума відстаней між кінцевим числом точок, що утворюють цикл. Сума  $n$ -мірних двійкових векторів, що задають координати точок циклічної фігури дорівнює нулю. Дане визначення метрики ставить в основу не елементи множини, але відношення, що дозволяє скоротити систему аксіом із трьох до однієї й поширити її дію на як завгодно складні конструкції  $n$ -мірного простору. Класичне завдання метрики для визначення взаємодії однієї, двох і трьох точок у векторному логічному просторі, є окремим випадком  $\beta$ -метрики при  $i = 1, 2, 3$  відповідно:

$$M = \begin{cases} d_1 = 0 \leftrightarrow a = b; \\ d_1 \oplus d_2 = 0 \leftrightarrow d(a, b) = d(b, a); \\ d_1 \oplus d_2 \oplus d_3 = 0 \leftrightarrow d(a, b) \oplus d(b, c) = d(a, c). \end{cases}$$

Метрика  $\beta$  векторного логічного багатозначного простору є рівна порожній множині симетрична різниця відстаней між кінцевим числом точок, що утворюють цикл:

$$\beta = \bigwedge_{i=1}^n d_i = \emptyset.$$

Рівність порожній множині симетричної різниці теоретико-множинної взаємодії підкреслює рівнозначність компонентів, що беруть участь у формуванні рівняння, де єдина операція, використувувана в аналітичній (чотиризначної) моделі, визначається у вигляді:

$\Delta$	0	1	x	$\emptyset$
0	$\emptyset$	x	1	0
1	x	$\emptyset$	0	1
x	1	0	$\emptyset$	x
$\emptyset$	0	1	x	$\emptyset$

Для її визначення необхідно виконати нескладні перетворення, подано у вигляді наступних доведенняів:

$$\begin{aligned} x\Delta 0 &= (x \cap \tilde{0}) \cup (\tilde{x} \cap 0) = (x \cap 1) \cup (\emptyset \cap 0) = 1; \\ x\Delta 1 &= (x \cap \tilde{1}) \cup (\tilde{x} \cap 1) = (x \cap 0) \cup (\emptyset \cap 1) = 0; \\ \emptyset\Delta 0 &= (\emptyset \cap \tilde{0}) \cup (\tilde{\emptyset} \cap 0) = (\emptyset \cap 1) \cup (x \cap 0) = 0; \\ \emptyset\Delta 1 &= (\emptyset \cap \tilde{1}) \cup (\tilde{\emptyset} \cap 1) = (\emptyset \cap 0) \cup (x \cap 1) = 1; \\ 0\Delta 1 &= (0 \cap \tilde{1}) \cup (\tilde{0} \cap 1) = (0 \cap 0) \cup (1 \cap 1) = x; \\ x\Delta \emptyset &= (x \cap \tilde{\emptyset}) \cup (\tilde{x} \cap \emptyset) = (x \cap x) \cup (\emptyset \cap \emptyset) = x; \\ x\Delta x &= (x \cap \tilde{x}) \cup (\tilde{x} \cap \emptyset) = (\emptyset \cap x) \cup (\emptyset \cap \emptyset) = \emptyset. \end{aligned}$$

При переході до двозначного алфавіту модель синтезу або аналізу кібернетичної системи трансформується у вигляді

$$f(F, T, L) = 0 \rightarrow F \oplus T \oplus L = 0,$$

де фігурує вже симетрична хог-операція, що має таблицю істинності:

$\oplus$	0	1
0	0	1
1	1	0

### 3.2 Синтез тестів $F \oplus L$ -методом

Використовується рівняння простору  $f(F, T, L) = 0 \rightarrow F \oplus T \oplus L = 0$ , що трансформується до виду  $T = F \oplus L$ . Наприклад, для функціональності логічного ог-елемента синтез тесту можна представити у вигляді T-процедури:

$$T = (F \oplus L) \cap F,$$

де  $L$  є матриця одиночних дефектів змінних логічного елемента, що задає можливість транспортування дефектів від вхідних ліній до виходу, а  $F$  - таблиця істинності:

$$F = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array}; L = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ 0 & 1 & 1 \\ \hline \end{array}$$

Виконання T-процедури пов'язане із застосуванням хог-операції між векторами F і L. Кількість векторних операцій дорівнює добутку потужностей векторів у кожній таблиці:

$$Q(T) = (\text{card}F \times \text{card}L) \times \text{card}F = \text{card}F^2 \times \text{card}L.$$

Для og-елемента T-процедура має такий вигляд:

$$\begin{array}{c}
 \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ \hline \end{array} \rightarrow T_i \in F \rightarrow \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ 0 & 0 & 0 \\ \hline \end{array} \\
 \\
 \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ \hline \end{array} \rightarrow T_i \in F \rightarrow \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 1 & 1 \\ 0 & 0 & 0 \\ \hline \end{array}
 \end{array} \left. \vphantom{\begin{array}{c} \\ \\ \end{array}} \right\} \cup \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ \hline \end{array}$$

Якщо функціональність задана у вигляді кубічного покриття в трійчному алфавіті, то для синтезу тестів необхідно використовувати несиметричну хог-операцію, що має вигляд:

$\tilde{\oplus}$	0	1
0	0	1
1	1	0
x	1	0

Для кубічного покриття and-елемента T-процедура  $T = (F \tilde{\oplus} L) \cap F$  має такий вигляд:

$$\left. \begin{array}{l}
 \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & X & 0 \\ X & 0 & 0 \\ 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ \hline \end{array} \rightarrow T_i \in F \rightarrow \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 0 & 1 & 0 \\ \hline \end{array} \\
 \\
 \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & X & 0 \\ X & 0 & 0 \\ 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ \hline \end{array} \rightarrow T_i \in F \rightarrow \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 1 & 0 & 0 \\ \hline \end{array}
 \end{array} \right\} \cup \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline \end{array}$$

У загальному випадку виконання Т-процедури для довільної функціональності, представленої кубічним покриттям, показано в наступному прикладі:

$$\begin{array}{l}
 \left. \begin{array}{l}
 \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline X & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ X & 0 & 1 & 0 \\ X & 1 & 0 & 0 \\ 0 & 1 & X & 0 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ \hline \end{array} \cup \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline \end{array} \cup \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ \hline \end{array} \right\} = \\
 = \begin{array}{l}
 \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ \hline \end{array} \cup \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ \hline \end{array} \cup \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ \hline \end{array} \right\} = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \hline \end{array}
 \end{array}$$

Після синтезу тестів необхідно використовувати тест і матрицю дефектів для побудови компактної таблиці несправностей  $\bar{F} = (T \bar{\oplus} L)$ , що призначена для виконання процедури пошуку дефектів:

$$\begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline \end{array} \bar{\oplus} \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline \bar{X}_1 & \bar{X}_2 & \bar{Y} \\ \hline \bar{1} & \bar{1} & \bar{1} \\ \bar{0} & \bar{1} & \bar{0} \\ \bar{1} & \bar{0} & \bar{0} \\ \hline \end{array}$$

У цьому випадку отримана таблиця  $\bar{F} = (T \bar{\oplus} L)$  є підмножину функціональності  $\bar{F} \in F = (T \bar{\oplus} L)$ , окремі координати якої відзначені символом інверсії, що означає перевіряємість несправності, інверсної стосовно справного

стану. Формально синтез таблиці несправностей використовує некомутативну операцію стиску двох битів даних, представлену в наступному вигляді:

$\oplus$	0	1
0	0	$\bar{0}$
1	1	$\bar{1}$

Нижче представлений варіант синтезу таблиці несправностей для більше складної функціональності (точками позначені нулі):

$X_1$	$X_2$	$X_3$	$Y$	$\oplus$	$X_1$	$X_2$	$X_3$	$Y$	$=$	$X_1$	$X_2$	$X_3$	$Y$
0	1	1	0		1	.	.	1		$\bar{0}$	$\bar{1}$	$\bar{1}$	$\bar{0}$
1	1	1	1		1	1	1	1		$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$
1	1	0	0		.	1	1	1		1	$\bar{1}$	0	$\bar{0}$
1	0	1	0		.	1	.	1		1	$\bar{0}$	$\bar{1}$	$\bar{0}$
1	0	0	1		.	1	1	1		1	$\bar{0}$	0	$\bar{1}$

Компактна таблиця несправностей є більше кращою для візуального оцінювання результатів взаємодії тесту й дефектів, що перевіряються на кожному вхідному наборі. При цьому виграш в обсязі пам'яті для зберігання таблиці відсутній.

### 3.3 Апарат булевих похідних для синтезу тестів

Розглядаються методи узяття булевих похідних по таблиці істинності, диз'юнктивній формі або кубічному покриттю для створення умов активізації вхідних змінних при синтезі тестів. Дослідження методу пропонується виконати за допомогою наступних трьох прикладів логічних функцій:



$$1) f(x) = x_1 \vee x_1 \bar{x}_2. \quad 2) f(x) = x_1 x_2 \vee \bar{x}_1 x_3. \quad 3) f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3.$$

Питання, що підлягають розв'язанню: 1) Визначення всіх похідних першого порядку по аналітичній, кубічній і табличній формі завдання логічної функції. 2) Верифікація отриманих умов активізації шляхом їхнього моделювання на одній з форм опису функціональності. 3) Синтез тестів активізації змінних логічної функції на основі обчислення похідних.

Приклад 3.1. Визначити всі похідні першого порядку за аналітичною формою логічної функції  $f(x) = x_1 \vee x_1 \bar{x}_2$ .

Застосування формули обчислення

$$f'(x_i) = \frac{df(x_1, x_2, \dots, x_i, \dots, x_n)}{dx_i} = f(x_1, x_2, \dots, x_i = 0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_i = 1, \dots, x_n)$$

визначає булеву похідну першого порядку як суму по модулі дві нульові й одиничної залишкові функції.

Для розглянутої функції виходить:

$$\frac{df(x_1, x_2)}{dx_1} = f(0, x_2) \oplus f(1, x_2) = (0 \vee 0\bar{x}_2) \oplus (1 \vee 1\bar{x}_2) = 0 \oplus 1 = 1;$$

$$\frac{df(x_1, x_2)}{dx_2} = f(x_1, 0) \oplus f(x_1, 1) =$$

$$= (x_1 \vee x_1 \cdot \bar{0}) \oplus (x_1 \vee x_1 \cdot \bar{1}) = (x_1 \vee x_1 \cdot 1) \oplus (x_1 \vee x_1 \cdot 0) = (x_1 \vee x_1) \oplus (x_1 \vee 0) = x_1 \oplus x_1 = 0.$$

Нульове значення похідної означає відсутність умов активізації змінної  $x_2$ , що дає підстави вважати її несуттєвою, а, отже, забрати із числа змінна, формуюча функціональність.

Приклад 3.2. Визначити всі похідні першого порядку за аналітичною формою логічної функції  $f(x) = x_1 x_2 \vee \bar{x}_1 x_3$ . Для розглянутої функції виконуються наступні обчислення:

$$\begin{aligned} \frac{df(x_1, x_2, x_3)}{dx_1} &= f(0, x_2, x_3) \oplus f(1, x_2, x_3) = \\ &= (0 \cdot x_2 \vee \bar{0} \cdot x_3) \oplus (1 \cdot x_2 \vee \bar{1} \cdot x_3) = (0 \vee 1 \cdot x_3) \oplus (x_2 \vee 0 \cdot x_3) = x_3 \oplus x_2 = x_2 \bar{x}_3 \vee \bar{x}_2 x_3; \\ \frac{df(x_1, x_2, x_3)}{dx_2} &= f(x_1, 0, x_3) \oplus f(x_1, 1, x_3) = \bar{x}_1 x_3 \oplus (x_1 \vee x_3) = \\ &= \bar{x}_1 x_3 (x_1 \vee x_3) \vee \overline{\bar{x}_1 x_3 (x_1 \vee x_3)} = (x_1 \vee \bar{x}_3)(x_1 \vee x_3) \vee \bar{x}_1 x_3 \bar{x}_1 \bar{x}_3 = x_1; \\ \frac{df(x_1, x_2, x_3)}{dx_3} &= f(x_1, x_2, 0) \oplus f(x_1, x_2, 1) = x_1 x_2 \oplus (\bar{x}_1 \vee x_2) = \\ &= \overline{x_1 x_2 (\bar{x}_1 \vee x_2)} \vee x_1 x_2 (\bar{x}_1 \vee x_2) = (\bar{x}_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2) \vee x_1 x_2 x_1 \bar{x}_2 = \bar{x}_1. \end{aligned}$$

Для трьох змінних отримані 4 умови активізації, які відповідають чотирьом логічним шляхам у схемній структурі диз'юнктивної форми даної функції.

Приклад 3.3. Визначити всі похідні першого порядку за кубічною формою логічної функції

$$f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3 =$$

$x_1$	$x_2$	$x_3$	$Y$
X	0	0	1
1	1	1	1
X	0	1	0
X	1	0	0
0	1	X	0

$$=$$

$x_2 x_3$				
$x_1$	00	01	11	10
0	1	0	0	0
1	1	0	1	0

$$=$$

$x_1$	$x_2$	$x_3$	$Y$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Процес-модель обчислення похідної за змінною  $x_i$  для функції, заданою табличною формою, має наступні пункти 1) Моделювання за таблицею

істинності (кубічному покриттю) вхідних наборів, для визначення стовпця  $Y_i^0$ , де змінна  $x_i$ , має тільки нульове значення для всіх рядків таблиці істинності. Число таких наборів завжди дорівнює  $q = 2^{n-1}$ ,  $n$  – число змінних. 2) Обчислення координат стовпця  $Y_i^1$ , з одиничним значенням змінної  $x_i$  для всіх рядків таблиці. 3) Обчислення стовпця  $Y_i^\oplus = Y_i^0 \oplus Y_i^1$  з урахуванням правила  $0 \oplus X \vee 1 \oplus X = X$ . 4) Формування диз'юнктивної форми похідної функції за одиничним значенням стовпця  $Y_i^\oplus$  без змінної  $x_i$ , по якій береться похідна. Інакше, фіксуються рядки таблиці, що відповідають одиничним значенням стовпця  $Y_i^\oplus$ , що визначає похідну функції. Аналітична модель процесу узяття похідній по функції, представленою таблицею, має такий вигляд:

$$\frac{df}{dx_i} = f(x_1, x_2, \dots, x_i = 0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_i = 1, \dots, x_n);$$

$$Y_i^\oplus = [Y_i^0 = f(x_1, x_2, \dots, x_i = 0, \dots, x_n)] \oplus [Y_i^1 = f(x_1, x_2, \dots, x_i = 1, \dots, x_n)].$$

Для двох різних табличних форм результат обчислення похідній за першою змінною зображено нижче:

$$\frac{df}{dx_1} = \begin{array}{|c|c|c|c|c|c|c|} \hline x_1 & x_2 & x_3 & Y & Y_1^0 & Y_1^1 & Y_1^\oplus \\ \hline X & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline X & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline X & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & X & 0 & 0 & X & 1 \\ \hline \end{array} = x_2 \vee x_2 x_3 = x_2 x_3;$$

$$\frac{df}{dx_1} = \begin{array}{|c|c|c|c|c|c|c|} \hline x_1 & x_2 & x_3 & Y & Y_1^0 & Y_1^1 & Y_1^\oplus \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline \end{array} = x_2 x_3.$$

При неоднозначному значенні похідної для функції, заданої кубічним покриттям, вибирається терм, що має максимальне число змінних. Мінімізація похідної функції на основі тотожності  $a \vee ab = a$  не зберігає умови активізації змінної, по якій береться похідна. Справді, значення функції від трьох змінних при умовах  $a\bar{b} \vee ab$  може бути дорівнює нулю (одиниці), що означає можливість відсутності зміни функції при активізації змінної «з» (останній стовпець карти Карно):

$\frac{ab}{c}$	00	01	11	10
0	1	0	0	0
1	1	0	1	0

Лема неперетинаємості кубів. Можливість коректного узяття похідній для одержання тесту активізації по змінній  $x_i$  обмежується такою мінімальною структурою кубічного покриття або аналітичної диз'юнктивної (кон'юнктивної) нормальною формою, де перетинання будь-яких кубів (рядків таблиці істинності) або термів ДНФ (КНФ) дає порожня множина:

$$\frac{df}{dx_i} = f(x_1, x_2, \dots, x_i = 0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_i = 1, \dots, x_n) \in T \leftrightarrow \forall i, j (C_i \cap C_j = \emptyset); i, j = \overline{1, n}; i \neq j.$$

Справді, якщо покриття, представлене вище, записати за правилами неперетинаємості кубів, те всі похідні будуть валідні для синтезу тестів без додаткової перевірки:

$$f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3 =$$

x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	Y
X	0	0	1
1	1	1	1
X	0	1	0
X	1	0	0
0	1	X	0

 $\rightarrow$ 

x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	Y
X	0	0	1
1	1	1	1
X	0	1	0
X	1	0	0
0	1	1	0

Щоб одержати таке кубічне покриття, необхідно виконувати мінімізацію всіма існуючими методами (карти Карно, Квайна, істотних змінних, невизначених коефіцієнтів, бінарного графа) з обліком правила: покриття нульових і одиничних координат таблиці істинності в процесі мінімізації не повинні перетинатися. У розглянутому випадку, коли функціональність переписана з урахуванням даного правила, навіть загальне число кубів не змінилося, у той час як покриття придбало якість неперетинаємості (як у таблиці істинності) для синтезу тестів активізації змінних:

$$\frac{df}{dx_1} =$$

x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	Y	Y <sub>2</sub> <sup>0</sup>	Y <sub>2</sub> <sup>1</sup>	Y <sub>2</sub> <sup>⊕</sup>
X	0	0	1	1	1	0
1	1	1	1	0	1	1
X	0	1	0	0	0	0
X	1	0	0	0	0	0
0	1	1	0	0	0	0

 $= x_2 x_3; \frac{df}{dx_1} =$ 

x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	Y	Y <sub>2</sub> <sup>0</sup>	Y <sub>2</sub> <sup>1</sup>	Y <sub>2</sub> <sup>⊕</sup>
0	0	0	1	1	1	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	1	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	1	0	1	1

 $= x_2 x_3.$

Обчислення похідних за всіма вхідними змінними дає можливість побудувати тест активізації для функціональності, заданої вже не таблицею істинності, а кубічним покриттям, що може істотно зменшити час синтезу тестів. Для другої змінної функції  $f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3$  процес обчислення похідних для трьох різних форм (кубічної, табличної й аналітичної) має такий вигляд:

$x_1$	$x_2$	$x_3$	$Y$	$Y_2^0$	$Y_2^1$	$Y_2^\oplus$
X	0	0	1	1	0	1
1	1	1	1	0	1	1
X	0	1	0	0	X	X
X	1	0	0	1	0	1
0	1	1	0	0	0	0

$$= \bar{x}_3 \vee x_1 x_3 \vee \bar{x}_3 = \bar{x}_3 \vee x_1 x_3;$$

$x_1$	$x_2$	$x_3$	$Y$	$Y_2^0$	$Y_2^1$	$Y_2^\oplus$
0	0	0	1	1	0	1
0	0	1	0	0	0	0
0	1	0	0	1	0	1
0	1	1	0	0	0	0
1	0	0	1	1	0	1
1	0	1	0	0	1	1
1	1	0	0	1	0	1
1	1	1	1	0	1	1

$$= \vee \bar{x}_1 \bar{x}_3 \vee x_1 \bar{x}_3 \vee x_1 x_3 = \bar{x}_3 \vee x_1 x_3;$$

Аналогічний результат отриманий шляхом визначення похідної за диз'юнктивною нормальною формою логічної функції:

$$\begin{aligned} \frac{df(x_1, x_2, x_3)}{dx_2} &= f(x_1, 0, x_3) \oplus f(x_1, 1, x_3) = (\bar{x}_3) \oplus (x_1 x_3) = (x_3) (x_1 x_3) \vee (\bar{x}_3) (\overline{x_1 x_3}) = \\ &= (x_1 x_3) \vee (\bar{x}_1 \bar{x}_3 \vee \bar{x}_3 \bar{x}_3) = x_1 x_3 \vee \bar{x}_3. \end{aligned}$$

Для третьої змінної функції  $f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3$  похідні від трьох різних форм (кубічної, табличної й аналітичної) представлені в наступному виді:

$$\frac{df}{dx_3} = \begin{array}{c|ccc|ccc} & x_1 & x_2 & x_3 & Y & Y_3^0 & Y_3^1 & Y_3^\oplus \\ \hline X & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ X & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ X & 1 & 0 & 0 & 0 & 0 & X & X \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} = \bar{x}_2 \vee x_1 x_2;$$

$$\begin{array}{c|ccc|ccc} & x_1 & x_2 & x_3 & Y & Y_3^0 & Y_3^1 & Y_3^\oplus \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{array} = \sqrt{\bar{x}_1 \bar{x}_2} \vee x_1 \bar{x}_2 \vee x_1 x_2 = \bar{x}_2 \vee x_1 x_2.$$

$$\begin{aligned} \frac{df(\bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3)}{dx_3} &= f(x_1, x_2, 0) \oplus f(x_1, x_2, 1) = (\bar{x}_2) \oplus (x_1 x_2) = (x_2) (x_1 x_2) \vee (\bar{x}_2) \overline{(x_1 x_2)} = \\ &= (x_1 x_2) \vee (\bar{x}_1 \bar{x}_2 \vee \bar{x}_2 \bar{x}_2) = x_1 x_2 \vee \bar{x}_2. \end{aligned}$$

Таким чином, всі результати по обчисленню похідних від трьох форм завдання функції є ідентичними. Найбільше технологічним є метод обчислення похідної за таблицею істинності. Але використання кубічного покриття має меншу обчислювальну складність у силу компактного подання функціональності за рахунок введення надмірності (символу X) у двійковий алфавіт. Використання аналітичної форми припускає істотне підвищення складності алгоритмів, пов'язаної із застосуванням законів булевої алгебри й мінімізації функцій, що обмежує її застосування для розв'язок практичних завдань.

Процес-модель одержання тесту  $T = [T_{ij}]$ ,  $i = \overline{1, k}$ ;  $j = \overline{1, n}$  комбінаційної функціональності:

- 1)  $f'(x_i) = f(x_1, x_2, \dots, x_i = 0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_i = 1, \dots, x_n)$ ;
- 2)  $T = \bigcup_{i=1}^n [f'(x_i) * (x_i = 0) \vee (x_i = 1)]$ ;
- 3)  $T_{ij} = T_{i-1,j} \leftarrow T_{ij} = X; T_{1j} = 1 \leftarrow T_{1j} = X$ ;
- 4)  $T = T \setminus T_i \leftarrow T_i = T_{i-r}, r = \overline{1, i-1}, i = \overline{2, n}$ .

1) Обчислення похідних за всіма  $n$  змінним функціональності шляхом використання однієї з форм: аналітичної табличної, кубічної. 2) Об'єднання всіх умов (векторів) активізації в таблицю, де кожному вектору шляхом конкатенації (\*) ставиться у відповідність зміна змінної, по якій була взята похідна, що означає подвоєння числа тестових наборів стосовно загальної кількості ( $k$ ) умов активізації. 3) Довизначення символу  $X=\{0,1\}$  у координаті шляхом присвоєння двійкового значення однойменної координати в попередньому векторі для одержання тесту мінімальної довжини. 4) Мінімізація тестових векторів шляхом видалення повторюваних вхідних послідовностей.

Рис. 3.2 ілюструє таблиці процесу одержання тесту відповідно до пунктів 2-4 алгоритми для функціональності  $f = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3$ , представленою схемною структурою.

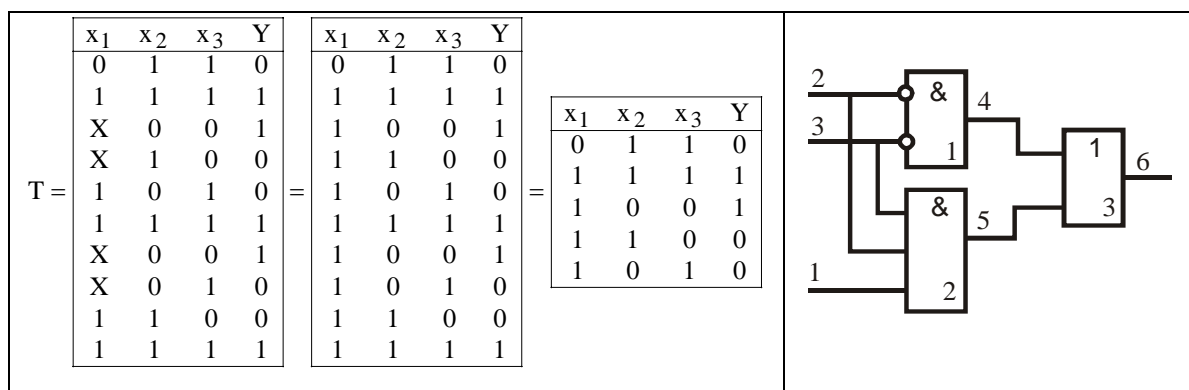


Рисунок 3.2 – Таблиці тестів і схемна структура булевої функції



Отриманий тест за кількістю і якістю ідентичний вхідним наборам, синтезованим раніше за допомогою  $F \oplus L$ -методу, отже, він має однакові властивості по покриттю несправностей і глибині пошуку дефектів.

Запропонована процес-модель синтезу тестів для функціональностей може бути використаний як убудований компонент інфраструктури сервісного обслуговування цифрових систем на кристалах з використанням стандарту граничного сканування IEEE 1500.

### 3.4 Векторно-логічні методи аналізу таблиць несправностей

Аналітична модель верифікації HDL-коду з використанням механізму темпоральних асерцій орієнтована на досягнення заданої глибини діагностування й представлена в наступному вигляді:

$$M = f(F, L, T, C, A, t); C = \{C_1, C_2, \dots, C_i, \dots, C_m\}; L = \{L_1, L_2, \dots, L_i, \dots, L_n\}; \\ A(t) = \{A_1, A_2, \dots, A_i, \dots, A_k\}; A \subseteq L; F = L \times C; k \leq n; T = \{T_1, T_2, \dots, T_i, \dots, T_p\}.$$

Тут  $C_i$  – група операторів коду, навантажена на вершину  $L_i$  (змінна, регістр, лічильник, пам'ять) і формуючий її стан;  $F$  – функціональність, представлена транзакційним графом  $F = L \times C$  у вигляді декартова добутку множини вершин і дуг;  $A$  – сукупність темпоральних асерцій, як підмножина вершин транзакційного графа  $A \subseteq L$ . Метод пошуку функціональних порушень (ФН) блоку операторів коду використовує попередньо побудовану таблицю ФН  $V = [V_{ij}]$ , де рядок є відношення між тестовим сегментом і підмножиною програмних блоків  $T_i \approx (V_{i1}, V_{i2}, \dots, V_{ij}, \dots, V_{in})$  з можливими ФН. Стовець таблиці формує відношення між програмним блоком і тестовими сегментами  $V_j \approx (T_{1j}, T_{2j}, \dots, T_{ij}, \dots, T_{pj})$ , які можуть перевіряти блок із ФН. На стадії моделювання визначається узагальнена реакція  $m = \{m_1, m_2, \dots, m_i, \dots, m_p\}$  механізму асерцій  $F$  на тест, шляхом формування

$m_i = (A_1 \vee A_2 \vee \dots \vee A_i \vee \dots \vee A_k)$ ,  $A_i = \{0,1\}$  як реакції асерцій на тест-сегмент  $T_i$ . Пошук ФН заснований на визначенні хог-операції між вектором стану асерцій і стовпців таблиці ФН  $m \oplus (B_1 \vee B_2 \vee \dots \vee B_j \vee \dots \vee B_n)$ . Вибір розв'язок визначається сукупністю векторів  $B_j$  з мінімальним числом одиничних координат

$$B = \min_{j=1,n} [B_j = \sum_{i=1}^p (B_{ij} \oplus m_i)],$$

формуючі програмні блоки із ФН, що перевіряються на тестових сегментах.

Підвищення адекватності моделі діагностування ФН блоків HDL-Коду пов'язане з розширенням простору реакції проекту в процесі його тестової верифікації. У цьому випадку варто розширити простір існування механізму асерцій до двовимірного шляхом модифікації векторів  $m$  і  $A$ :

$$A_i = (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{ik}); m_i = (m_{i1}, m_{i2}, \dots, m_{ij}, \dots, m_{ik});$$

$$B_{ij} = (B_{ij1}, B_{ij2}, \dots, B_{ijr}, \dots, B_{ijk}); \{A_{ij}, m_{ij}, B_{ijr}\} = \{0,1\}; i = \overline{1, p}; j = \overline{1, n}; r = \overline{1, k}.$$

При цьому таблиця ФН стає тривимірної по параметрах:  $i = \overline{1, p}$ ;  $j = \overline{1, n}$ ;  $r = \overline{1, k}$  – числа рядків або сегментів-тест-сегментів, стовпців або функціональних блоків HDL-моделі, темпоральних асерцій або точок спостереження для HDL-коду відповідно:

B	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>	A <sub>8</sub>	m
	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	B <sub>8</sub>	
T <sub>1</sub>	1001	1110	0001	0110	0011	1110	0001	1110	1001
T <sub>2</sub>	1010	1000	0001	0110	0001	0001	1110	1110	1000
T <sub>3</sub>	1110	0001	1111	0111	1011	1110	1001	1101	0011
T <sub>4</sub>	1000	0111	0110	1001	1000	0001	1011	1110	1011
T <sub>5</sub>	0001	0110	1100	0111	0010	1110	0001	1000	1001
T <sub>6</sub>	1110	1011	0110	0001	1110	1100	1001	0111	1110
T <sub>7</sub>	0110	1101	0110	1000	0111	1011	0101	1110	0001
T <sub>8</sub>	1000	1110	1001	0100	0101	1001	0111	0001	1110

Потенційно тривимірність таблиці ФН збільшує обсяг діагностичної інформації, що дає можливість підвищити глибину пошуку некоректного програмного блоку, у межі до оператора або рядка коду. Однак для аналізу тривимірної таблиці ФН необхідно модифікувати запропоновані методи діагностування.

Пошук дефектів за таблицею несправностей або функціональних порушень на основі вектора експериментальної перевірки – реакції (m) функціональності на тест  $m = (m_1, m_2, \dots, m_i, \dots, m_n), m_i = \{0, 1\}$  зводиться до методів аналізу рядків або стовпців.

Перший метод заснований на застосуванні векторної хог-операції між реакцією m функціональності на тест, формально розглянутого в якості вхідний вектор-стовпця або маски m, і стовпців таблиці несправностей  $m \oplus (B_1 \vee B_2 \vee \dots \vee B_j \vee \dots \vee B_m)$ . Для підрахунку якості взаємодії векторів  $Q_j(m \oplus B_j)$  з метою вибору кращого розв'язок визначаються стовпці с мінімальним числом одиниць результуючого вектора

$$B = \min_{j=1,k} [B_j = \sum_{i=1}^n (B_{ij} \oplus m_i)].$$

Вони ідентифікують і формують дефектні блоки з функціональними порушеннями, що перевіряються на тестових наборах. Аналітична модель процесу одержання розв'язок у вигляді списку блоків з функціональними несправностями, присутніх у програмному продукті, представлена в наступному вигляді:

$$B = \min_{j=1,k} [B_j = \sum_{i=1}^n (B_{ij} \oplus m_i)]; \quad B^0 = \forall_{j=1,k} [B_j = \sum_{i=1}^n (B_{ij} \oplus m_i) = 0]. \quad (3.1)$$

Тут фігурує вектор експериментальної перевірки, що є вхідним для наступного аналізу таблиці ФН

$$m = f(A, B) \oplus f^*(A, B, L) \quad (3.2)$$

є результат проведення тестового експерименту – порівняння функціоналів (станів виходів) еталонного  $f(A, B)$  й реального  $f^*(A, B, L)$  пристрою з дефектами  $L$  на тестових наборах  $A$ . У другому випадку, якщо множина  $B^0 > 1$ , це означає наявність еквівалентних, не помітних на даному тесті й механізмі асерцій, функціональних порушень. Вибір кращого розв'язку  $Y = \vee[(Q_1 \wedge Q_2) \oplus Q_1]$  з мінімальним числом одиничних координат із двох альтернатив, представлених векторними оцінками  $Q_1 \vee Q_2$ , здійснюється за допомогою схеми, зображеної на рис 3.3.

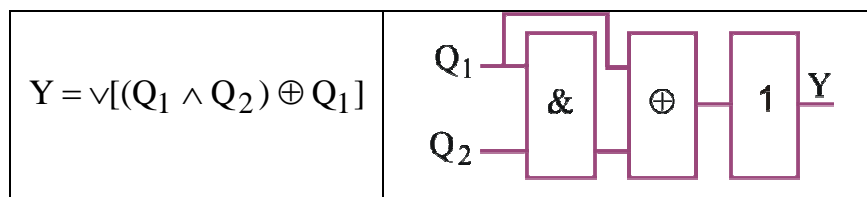


Рисунок 3.3 – Схема вибору кращого розв'язку

Перевага методу – вибір кращого розв’язку із всіх можливих одиночних і кратних ФН. По суті, у список дефектів включаються такі одиночні ФН, які при логічному множенні на вектор експериментальної перевірки дають результат у вигляді відповідного шпальти-колонки-стовпця-вектора-стовпця. Диз’юнкція всіх стовпців, що становлять розв’язок, дорівнює вектору експериментальної

$$\text{перевірки } \bigvee_{j=1}^r (B_j \in B) = m.$$

Обчислювальна складність методу аналізу стовпців визначається наступною залежністю:  $Z^c = 3n^2 + n^2 = 4n^2$ ;  $Z^r = 3n + n = 4n$ . Тут перша оцінка враховує виконання координатних операцій над матрицею, розмірністю  $n \times n$ . Друга оцінка визначає обчислювальну складність реєстрових паралельних операцій для підрахунку критеріїв якості й обробки матриці відповідно.

Логічний метод аналізу рядків таблиці несправностей. Стратегія визначення помилок програмного коду по таблиці ФН пов’язана з аналізом її рядків, що складають із двох процедур: 1) обчислення логічного добутку кон’юнкції рядків, відзначених одиничними значеннями вектора  $T_i (m_i = 1)$ , на заперечення диз’юнкції нульових рядків  $T_i (m_i = 0)$  для одиночних дефектних блоків; 2) обчислення логічного добутку диз’юнкції одиничних рядків на заперечення диз’юнкції нульових рядків для кратних дефектних блоків:

$$B^s = \left( \bigwedge_{\forall m_i=1} T_i \right) \wedge \overline{\left( \bigvee_{\forall m_i=0} T_i \right)}; \quad B^m = \left( \bigvee_{\forall m_i=1} T_i \right) \wedge \overline{\left( \bigvee_{\forall m_i=0} T_i \right)}. \quad (3.3)$$

Формули цікаві тим, що вони не прив’язані до критеріїв якості діагностування, а оперують лише двома компонентами, таблицею ФН і вектором експериментальної перевірки. Виконання процедури діагностування по формулах (3.3) для вектора експериментальної перевірки

$m_1 = (0101010010010)$ , заданого в останній таблиці ФН, дає результат:  $V^s(m_1, T) = D_2$ , що не гірше, ніж раніше отриманий методом аналізу стовпців. Для вектора експериментальної перевірки  $m_2 = (1110011100000)$  результат діагностування має вигляд:  $V^m(m_2, T) = L_1 \vee L_2$ . Обчислювальна складність методу аналізу рядків визначається наступною залежністю:  $Z^c = n^2$ ;  $Z^r = n$ . Перша оцінка призначена для підрахунку числа координатних операцій, друга визначає обчислювальну складність процесу обробки на основі реєстрових паралельних операцій. Запропоновані методи діагностування ФН для програмних і апаратних продуктів є один з найбільш істотних компонентів інфраструктури сервісного обслуговування проєктованих виробів.

### 3.5 3D-Метод діагностування дефектних компонентів у SoC

Мотивація визначається ринковою привабливістю матричного методу пошуку дефектів у компонентах (програмних і апаратних) цифрових систем на кристалах, як самого технологічного, що орієнтований на паралельну обробку даних, що дає можливість істотно зменшити час діагностичного обслуговування при виникненні несправностей.

Мета дослідження – створення моделі, методу і їхньої апаратної реалізації, орієнтованих на істотне зменшення часу тестування й витрат пам'яті для зберігання матриці діагностування шляхом формування тернарних відносин (тест-монітор-функціональний компонент).

Задачі: 1) Розробка моделі цифрової системи у вигляді транзакційного графа, а також матриці активізації функціональних компонентів на тестах щодо обраної множини моніторів. 2) Розробка методу аналізу матриці активізації для пошуку дефектів із заданою глибиною. 3) Синтез логічних функцій для убудованого діагностування дефектів.

Модель тестування цифрової системи представлена у вигляді наступного перетворення початкового рівняння діагнозу, певного хог-відношенням параметрів <тест - функціональність - несправні блоки>:

$$T \oplus F \oplus B = 0 \rightarrow B = T \oplus F \rightarrow B = \{T \times A\} \oplus F \rightarrow B = \{T \times A\} \oplus \{F \times m\},$$

яке оформлене в тернарне матричне відношення компонентів:

$$M = \{\{T \times A\} \times \{B\}\} \leftarrow M_{ij} = (T \times A)_i \oplus B_j.$$

Тут координата матриці (таблиці) дорівнює 1, якщо пари монітор-монітор-тест-монітор  $(T \times A)_i$  перевіряє (активізує) дефекти функціонального блоку  $B_j \in B$ .

Модель цифрової системи представлена у вигляді транзакційного графа  $G = \langle V, A \rangle$ ,  $V = \{V_1, V_2, \dots, V_i, \dots, V_n\}$ ,  $A = \{A_1, A_2, \dots, A_j, \dots, A_m\}$ , де визначені множина дуг – функціональних блоків і вершин – моніторів для спостереження сукупності змінних цифровий системи. Для цілей діагностування на графову модель накладається сукупність тестових сегментів  $T = \{T_1, T_2, \dots, T_r, \dots, T_k\}$ , що активізує транзакційні шляхи в графі. У загальному випадку модель тестування представлена декартовим добутком  $M = \langle V \times A \times T \rangle$ , що має розмірність  $Q = n \times m \times k$ . Щоб зменшити обсяг діагностичної інформації пропонується кожному тесту поставити у відповідність монітор, що відповідає за візуалізацію шляху активізації функціональних блоків, що дає можливість зменшити розмірність моделі (матриці) до  $Q = n \times k$  при збереженні всіх можливостей відносини тріади  $M = \langle V \times A \times T \rangle$ . Для пари монітор-монітор-тест-монітор можливі не тільки взаємо-однозначні відповідності  $\langle T_i \rightarrow A_j \rangle$ , але й функціональні  $\langle \{T_i, T_r\} \rightarrow A_j \rangle$ , а також ін'єктивні  $\langle T_i \rightarrow \{A_j, A_s\} \rangle$ . Таке

різноманіття відповідей дає можливість дублювати один тестовий сегмент для різних моніторів, так само як і навантажувати кілька тестів на той самий монітор. При цьому комірка матриці  $M_{ij} = \{0,1\}$  завжди зберігає свою розмірність, рівну одну битку.

Аналітична узагальнена модель матричного діагностування з використанням механізму моніторів орієнтована на досягнення заданої глибини пошуку дефектів і представлена в наступному вигляді:

$$M = f(G, L, T, B, A, t); B = \{B_1, B_2, \dots, B_i, \dots, B_m\}; L = \{L_1, L_2, \dots, L_i, \dots, L_n\}; \\ A(t) = \{A_1, A_2, \dots, A_i, \dots, A_k\}; A \subseteq L; G = L \times B; k \leq n; T = \{T_1, T_2, \dots, T_i, \dots, T_p\}.$$

Тут  $B_i$  – група операторів коду, навантажена на вершину  $L_i$  (змінна, реєстр, лічильник, пам'ять) і формуючий її стан;  $G$  – функціональність, представлена транзакційним графом  $G = (L, A) \times B$  у вигляді декартова добутку множини вершин і дуг;  $A$  – сукупність моніторів, як підмножина вершин транзакційного графа  $A \subseteq L$ . Метод пошуку несправностей функціональних блоків (ФБ) використовує попередньо побудовану таблицю (матрицю) активізації ТАФБ  $M = [M_{ij}]$ , де рядок є відношення між тестовим сегментом і підмножиною активізованих блоків  $T_i \rightarrow A_j \approx (M_{i1}, M_{i2}, \dots, M_{ij}, \dots, M_{in}), M_{ij} = \{0,1\}$ , спостережуваних на моніторі  $A_j$ . Стовець таблиці формує відношення між функціональним блоком, тестовими сегментами й моніторами  $M_j = B_j \approx f(T, A)$ . У механізм моніторів може бути уведений параметр модельного часу, що частково ускладнює матрицю активізації, указуючи часовий або модельний такт, на якому виконується моніторинг стану вершини або функціонального блоку на сегменті-тест-сегменті  $A_j = f(T_i, B_j, t_j)$ .

Для діагностування несправностей на стадії моделювання визначається узагальнена реакція (вектор-стовпець)  $m = \{m_1, m_2, \dots, m_i, \dots, m_p\}$  механізму



моніторів  $A$  на сегменти-тест-сегменти  $T$ , шляхом формування  $m_i = f(T_i, A_i)$ . Пошук несправного ФБ заснований на визначенні хог-операції між вектором стану асерцій і стовпців таблиці ФН  $m \oplus (M_1 \vee M_2 \vee \dots \vee M_j \vee \dots \vee M_n)$ . Вибір розв'язок визначається методом хог-аналізу стовпців, шляхом вибору сукупності векторів  $B_j$  з мінімальним числом одиничних координат

$$B = \min_{j=1,n} [B_j = \sum_{i=1}^p (B_{ij} \oplus m_i)],$$

формуючи функціональні блоки з несправностями, що перевіряються на тестових сегментах. На додаток до моделі матричного діагностування необхідно описати наступні важливі властивості матриці:

$$\begin{aligned} 1) M_i &= (T_i - A_j); & 2) \bigvee_{i=1}^m M_{ij} &\rightarrow \forall M_j = 1; \\ 3) M_{ij} \bigoplus_{j=1}^n M_{rj} &\neq M_{ij}; & 4) M_{ij} \bigoplus_{i=1}^k M_{ir} &\neq M_{ij}; \\ 5) \log_2 n \leq k &\leftrightarrow \log_2 |B| \leq |T|; & 6) B_j &= f(T, A) \rightarrow B \oplus T \oplus A = 0. \end{aligned}$$

Властивості означають: 1) Кожний рядок матриці є відповідність або підмножина декартова добутку (монітор-монітор-тест-монітор). 2) Диз'юнкція всіх рядків матриці дає вектор, дорівнює одиницям по всіх координатах. 3) Всі рядки матриці різні, що виключає тестову надмірність. 4) Всі стовпці матриці різні, що виключає існування еквівалентних несправностей. 5) Число рядків матриці повинне бути більше двійкового логарифма від числа стовпців, що визначає потенційну діагностуємість всіх блоків. 6) Функція діагностування блоку залежить від сукупного тесту й моніторів, які повинні бути мінімізовані без порушення діагнозопридатності.

Для пояснення працездатності моделі й методу далі розглянуті функціональності трьох модулів, що входять до складу цифрового фільтра Добеші. Першим є компонент Row\_buffer, для якого створений транзакційний граф на основі RTL-Моделі, рис. 3.4. Вершини представлені станами змінних і моніторів, відповідальних за вхідні у вершину транзакції або дуги, яким відповідають функціональні блоки.

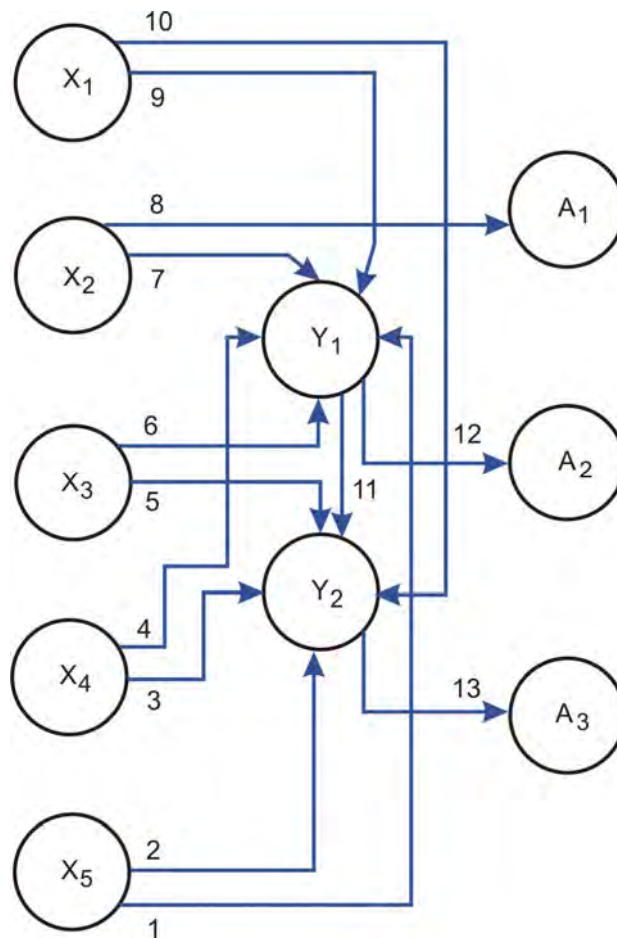


Рисунок 3.4 – Компонент Row\_buffer транзакційного графа

На основі графа, отриманого в процесі моделювання, будується таблиця активізації функціональних блоків, рядка якої представляють шляхи активізації блоків (T<sub>i</sub>) до замовленої вершини-монітору (A<sub>i</sub>). Таблиця являє собою покриття рядками-шляхами всіх стовпців або функціональних блоків. При

цьому в ній не повинне бути хоча б двох однакових стовпців. Відмінність таблиці полягає у формуванні пари <тест-тест-спостережувана вершина>, що дає можливість істотно скоротити розмірність таблиці при 100% розпізнаванні всіх дефектних блоків. Тут сама головна відмінність запропонованої моделі полягає в можливості опису за допомогою таблиці наступних відношень: різні тести – одна вершина; один тест – різні вершини:

A <sub>ij</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>7</sub>	T <sub>8</sub>	T <sub>9</sub>	T <sub>10</sub>	T <sub>11</sub>	T <sub>12</sub>	T <sub>13</sub>
t <sub>1</sub> → A <sub>3</sub>	.	.	.	.	.	.	.	1	.	.	.	.	.
t <sub>2</sub> → A <sub>1</sub>	1	.	.	.	.	.	.	.	.	.	.	.	1
t <sub>3</sub> → A <sub>1</sub>	.	.	1	.	.	.	.	.	.	.	.	.	1
t <sub>4</sub> → A <sub>1</sub>	.	.	.	.	1	.	.	.	.	.	.	.	1
t <sub>5</sub> → A <sub>1</sub>	.	.	.	.	.	.	.	.	.	.	1	.	1
t <sub>6</sub> → A <sub>1</sub>	.	.	.	.	.	.	.	.	1	.	.	.	1
t <sub>7</sub> → A <sub>2</sub>	.	1	.	.	.	.	.	.	.	.	.	1	.
t <sub>8</sub> → A <sub>2</sub>	.	.	.	1	.	.	.	.	.	.	.	1	.
t <sub>9</sub> → A <sub>2</sub>	.	.	.	.	.	1	.	.	.	.	.	1	.
t <sub>10</sub> → A <sub>2</sub>	.	.	.	.	.	.	1	.	.	.	.	1	.
t <sub>11</sub> → A <sub>2</sub>	.	.	.	.	.	.	.	.	.	1	.	1	.

За допомогою матриці активізації функціональних блоків (транзакційного графа) і хор-методу пошуку дефектів досить просто синтезувати логічні функції для формування комбінаційної схеми, що визначає в процесі й за результатами моделювання номер функціонального блоку, що має семантичні помилки:

$$A_3 = T_8^1;$$

$$A_1 = T_{13}^1 T_1^1 \vee T_{13}^1 T_3^1 \vee T_{13}^1 T_5^1 \vee T_{13}^1 T_{11}^1 \vee T_{13}^1 T_9^1;$$

$$A_2 = T_{12}^1 T_2^1 \vee T_{12}^1 T_4^1 \vee T_{12}^1 T_6^1 \vee T_{12}^1 T_7^1 \vee T_{12}^1 T_{10}^1;$$

Така властивість стає можливим завдяки відсутності еквівалентних несправностей або однакових стовпців у матриці активізації. Тому фіксація

фактичного стану всіх моніторів у вершинах  $A_1, A_2, A_3$  на 11 тестових наборах дає можливість однозначно ідентифікувати некоректний функціональний модуль шляхом виконання хог-операції між вектором асерцій і стовпцями матриці активізації. Нульове значення всіх координат результату хог-операції визначає номер стовпця, що відповідає несправному модулю. Імплементация моделі й методу в логічну функцію дає можливість визначати несправний блок ще до завершення діагностичного експерименту, якщо це можливо. Це означає істотну економію часу діагностування окремих видів дефектів. Наприклад, тест-монітор  $t_1 \rightarrow A_3$  дає можливість ідентифікувати вже на першому тесті несправність блоку  $T_8$ .

У якості другого тестового приклада для практичного використання розробленої моделі активізації й хог-методу пошуку дефектів далі пропонується синтез матриці діагностування для модуля дискретного косинусного перетворення з Xilinx бібліотеки, фрагмент якого представлений лістингом 3.1.

#### Лістинг 3.1. Фрагмент функціонального покриття

```
c0: coverpoint xin
{
  bins minus_big={{128:235}};
  bins minus_sm={{236:255}};
  bins plus_big={{21:127}};
  bins plus_sm={{1:20}};
  bins zero={0};
}
c1: coverpoint dct_2d
{
  bins minus_big={{128:235}};
  bins minus_sm={{236:255}};
  bins plus_big={{21:127}};
  bins plus_sm={{1:20}};
  bins zero={0};
  bins zero2=(0=>0);
}
endgroup
```

Для всіх 12 модулів фільтра розроблені транзакційні графи, таблиці активізації й логічні функції для тестування й пошуку дефектів дискретного косинусного перетворення. Дві графи з матрицями активізації й логічних функцій (3.5, 3.6), також приналежному фільтру, представлені нижче.

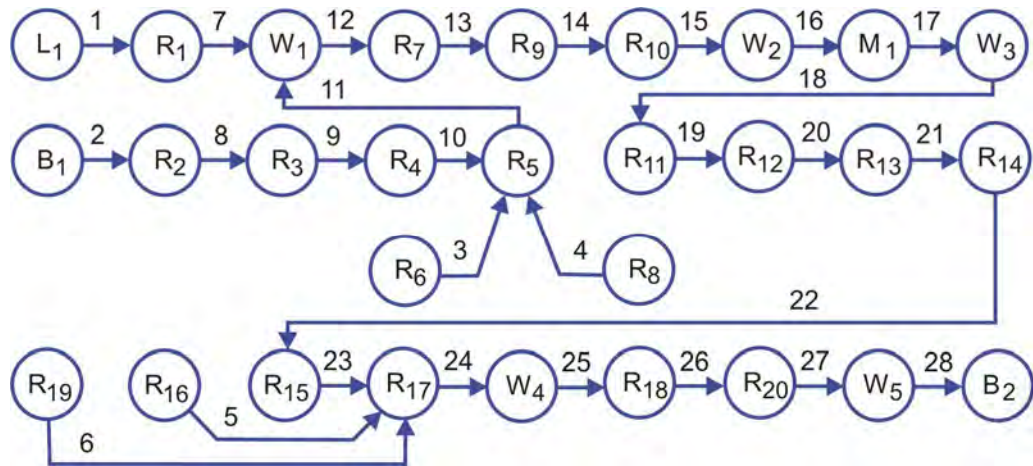


Рисунок 3.5 – Транзакційний граф Xilinx DWT-Модуля

Даному графові ставиться у відповідність наступна матриця активізації тестовий шлях (Pi) – асерція (Vi, Ri, Wi, Mi):

.	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>7</sub>	T <sub>8</sub>	T <sub>9</sub>	T <sub>10</sub>	T <sub>11</sub>	T <sub>12</sub>	T <sub>13</sub>	T <sub>14</sub>	T <sub>15</sub>	T <sub>16</sub>	T <sub>17</sub>	T <sub>18</sub>	T <sub>19</sub>	T <sub>20</sub>	T <sub>21</sub>	T <sub>22</sub>	T <sub>23</sub>	T <sub>24</sub>	T <sub>25</sub>	T <sub>26</sub>	T <sub>27</sub>	T <sub>28</sub>
P <sub>1</sub> → B <sub>2</sub>	1	.	.	.	.	.	1	.	.	.	.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P <sub>2</sub> → B <sub>2</sub>	.	1	.	.	.	.	.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P <sub>3</sub> → B <sub>2</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P <sub>4</sub> → B <sub>2</sub>	.	.	.	1	.	.	.	.	.	.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P <sub>5</sub> → B <sub>2</sub>	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	1	1	1
P <sub>6</sub> → B <sub>2</sub>	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	1	1	1
P <sub>2</sub> → R <sub>3</sub>	.	1	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
P <sub>2</sub> → R <sub>4</sub>	.	1	.	.	.	.	.	1	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
P <sub>3</sub> → R <sub>7</sub>	.	.	1	.	.	.	.	.	.	.	1	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
P <sub>3</sub> → R <sub>9</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
P <sub>3</sub> → R <sub>10</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.
P <sub>3</sub> → W <sub>2</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.	.
P <sub>3</sub> → M <sub>1</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.	.
P <sub>3</sub> → W <sub>3</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.
P <sub>3</sub> → R <sub>11</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.
P <sub>3</sub> → R <sub>12</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.
P <sub>3</sub> → R <sub>13</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	1	1	1	1	1	1	.	.	.	.	.	.	.	.
P <sub>3</sub> → R <sub>14</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	1	1	1	1	1	1	1	.	.	.	.	.	.	.
P <sub>3</sub> → R <sub>15</sub>	.	.	1	.	.	.	.	.	.	.	1	1	1	1	1	1	1	1	1	1	1	1	.	.	.	.	.	.
P <sub>5</sub> → W <sub>4</sub>	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.
P <sub>5</sub> → R <sub>18</sub>	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	.	.	.
P <sub>3</sub> → R <sub>20</sub>	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	1	.	.
P <sub>5</sub> → W <sub>5</sub>	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	1	1	.

Система функцій діагностування:

$$\begin{aligned}
 B_2 = & T_1 T_7 T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19} T_{20} T_{21} T_{22} T_{23} T_{24} T_{25} T_{26} T_{27} T_{28} \vee \\
 & \vee T_2 T_8 T_9 T_{10} T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19} T_{20} T_{21} T_{22} T_{23} T_{24} T_{25} T_{26} T_{27} T_{28} \vee \\
 & \vee T_3 T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19} T_{20} T_{21} T_{22} T_{23} T_{24} T_{25} T_{26} T_{27} T_{28} \vee \\
 & \vee T_4 T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19} T_{20} T_{21} T_{22} T_{23} T_{24} T_{25} T_{26} T_{27} T_{28} \vee \\
 & \vee T_5 T_{24} T_{25} T_{26} T_{27} T_{28} \vee T_6 T_{24} T_{25} T_{26} T_{27} T_{28};
 \end{aligned}$$

$$R_3 = T_2 T_8; R_4 = T_2 T_8 T_9; R_7 = T_3 T_{11} T_{12}; R_9 = T_3 T_{11} T_{12} T_{13};$$

$$R_{10} = T_3 T_{11} T_{12} T_{13} T_{14}; W_2 = T_3 T_{11} T_{12} T_{13} T_{14} T_{15}; M_1 = T_3 T_{11} T_{12} T_{13} T_{14} T_{15} T_{16};$$

$$W_3 = T_3 T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17}; R_{11} = T_3 T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18};$$

$$R_{12} = T_3 T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19}; R_{13} = T_3 T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19} T_{20};$$

$$R_{14} = T_3 T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19} T_{20} T_{21};$$

$$R_{15} = T_3 T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19} T_{20} T_{21} T_{22}; W_4 = T_5 T_{24}; R_{18} = T_5 T_{24} T_{25};$$

$$R_{20} = T_5 T_{24} T_{25} T_{26}; W_5 = T_5 T_{24} T_{25} T_{26} T_{27}.$$

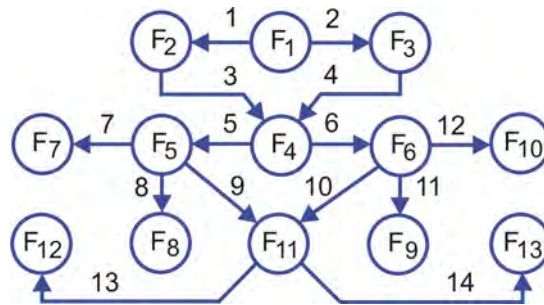


Рисунок 3.6 – Транзакційний граф main-RTL

Даному графові ставиться у відповідність наступна матриця діагностування тестовий шлях ( $P_i$ ) – асерція ( $F_i$ ):

$A_{ij}$	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$	$T_9$	$T_{10}$	$T_{11}$	$T_{12}$	$T_{13}$	$T_{14}$
$P_1 \rightarrow F_7$	1	.	1	.	1	.	1	.	.	.	.	.	.	.
$P_2 \rightarrow F_8$	.	1	.	1	1	.	.	1	.	.	.	.	.	.
$P_3 \rightarrow F_9$	1	.	1	.	.	1	.	.	.	.	1	.	.	.
$P_4 \rightarrow F_{10}$	.	1	.	1	.	1	.	.	.	.	.	1	.	.
$P_5 \rightarrow F_{12}$	1	.	1	.	1	.	.	.	1	.	.	.	1	.
$P_6 \rightarrow F_{13}$	.	1	.	1	.	1	.	.	.	1	.	.	.	1
$P_1 \rightarrow F_2$	1	.	.	.	.	.	.	.	.	.	.	.	.	.
$P_2 \rightarrow F_3$	.	1	.	.	.	.	.	.	.	.	.	.	.	.

Система функцій діагностування:

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

$$F_{10} = T_4^1 T_5^1 T_6^1 T_{12}^1; F_{12} = T_1^1 T_3^1 T_5^1 T_9^1 T_{13}^1; F_{13} = T_2^1 T_4^1 T_6^1 T_{10}^1 T_{14}^1;$$

$$F_2 = T_1^1; F_3 = T_2^1;$$

Фрагмент механізму моніторів представлений лістингом 3.2.

Лістинг 3.2. Фрагмент коду механізму моніторів

```
sequence first( reg[7:0] a, reg[7:0] b);
reg[7:0] d;
(!RST, d=a)
##7 (b==d);
endsequence
property f(a,b);
@(posedge CLK)
// disable iff(RST||$isunknown(a)) first(a,b);
!RST | => first(a,b);
```

```

endproperty
odin:assert property (f(xin,xa7_in))
// $display("Very good");
else $error("The end, xin =%b,xa7_in=%b", $past(xin, 7),xa7_in);

```

У результаті тестування дискретного косинусного перетворення в середовищі Riviera, Aldec були знайдені неточності в семи рядках HDL-моделі:

```
//add_sub1a <= xa7_reg + xa0_reg;//
```

Наступна корекція коду привела до лістингу 3.3.

Лістинг 3.3. Виправлений фрагмент коду

```

add_sub1a <= ({xa7_reg[8],xa7_reg} + {xa0_reg[8],xa0_reg});
add_sub2a <= ({xa6_reg[8],xa6_reg} + {xa1_reg[8],xa1_reg});
add_sub3a <= ({xa5_reg[8],xa5_reg} + {xa2_reg[8],xa2_reg});
add_sub4a <= ({xa4_reg[8],xa4_reg} + {xa3_reg[8],xa3_reg});
end
else if (toggle == 1'b0)
begin
add_sub1a <= ({xa7_reg[8],xa7_reg} - {xa0_reg[8],xa0_reg});
add_sub2a <= ({xa6_reg[8],xa6_reg} - {xa1_reg[8],xa1_reg});
add_sub3a <= ({xa5_reg[8],xa5_reg} - {xa2_reg[8],xa2_reg});
add_sub4a <= ({xa4_reg[8],xa4_reg} - {xa3_reg[8],xa3_reg});

```

### 3.6 Мультирівнева модель і метод (движок) діагностування цифрової системи

Представлена мультидеревом В, де кожна вершина є тривимірна таблиця активізації функціональних (апаратних) модулів, а дуги, що виходять із її, є переходи на нижній рівень деталізації при діагностуванні, коли заміна розглянутого несправного функціонального блоку занадто дорога:



$$B = [B_{ij}^{rs}], \text{ card}B = \sum_{r=1}^n \sum_{s=1}^{m_r} \sum_{j=1}^{k_{rs}} B_{ij}^{rs},$$

$n$  – число рівнів мультидерева діагностування;  $m_r$  – кількість функціональних блоків або компонентів на рівні  $r$ ;  $k_{rs}$  – число компонентів у таблиці  $B^{rs}$ ;  $B_{ij}^{rs} = \{0,1\}$  – компонент таблиці активізації, визначений сигналами перевірки (неперевірки) функціонального модуля тест-сегментом  $T_{i-A_i}$  щодо спостережуваного монітора  $A_i$ . Кожна вершина-таблиця має число вихідних долілиць дуг, дорівнює кількості функціональних блоків, діагностуємих (поданих) у таблиці активізації. Структура мультидерева, що відповідає моделі багаторівневого діагностування, представлена на рис 3.7.

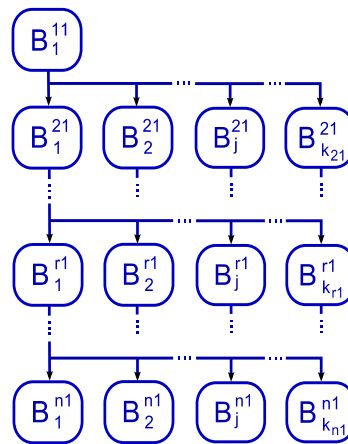


Рисунок 3.7 – Фрагмент мультидерева діагностування цифрових систем

Процес модель або метод пошуку дефектів за мультидеревом діагностування зводиться до створення движка, рис 3.8, для спуска по одній з гілок дерева на таку глибину, що задовольняє користувача за ступенем деталізації:

$$B_j^{rs} \oplus A^{rs} = \begin{cases} 0 \rightarrow \{B_j^{r+1,s}, R\}; \\ 1 \rightarrow \{B_{j+1}^{rs}, T\}. \end{cases}$$

Тут виконується векторна хог-операція між стовпцями матриці й вектором експериментальної перевірки  $A^{rs}$ , що визначається реакцією функціональності, знятої з моніторів (асерції або розряди регістра граничного сканування) при подачі всіх тест-сегментів.

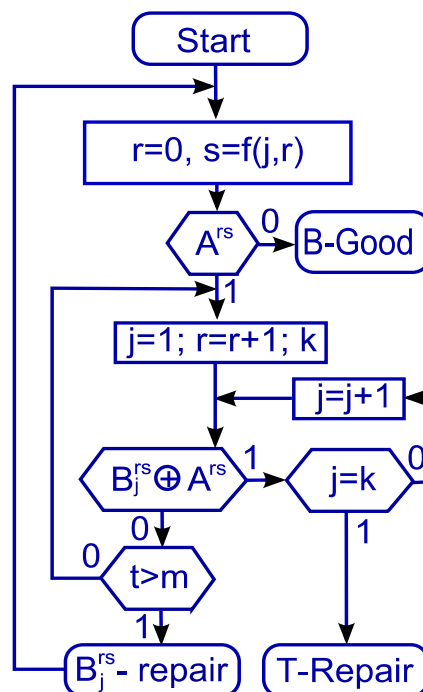


Рисунок 3.8 – Движок обходу мультидерева діагностування

Якщо хоча б одна координата отриманої векторної хог-суми дорівнює нулю  $B_j^{rs} \oplus A^{rs} = 0$ , то виконується одна з дій: перехід до матриці активізації нижнього рівня  $B_j^{r+1,s}$  або відновлення працездатності функціонального блоку  $B_j^{rs}$ . При цьому аналізується, що важливіше: 1) час – тоді виконується ремонт

розглянутого блоку з несправністю; 2) гроші – тоді здійснюється перехід долілиць, для уточнення місця дефекту, оскільки заміна більш дрібного блоку істотно зменшує вартість ремонту. Якщо хоча б одна координата отриманого вектора хог-суми дорівнює одиниці  $V_j^{rs} \oplus A^{rs} = 1$ , то виконується перехід до аналізу наступного стовпця матриці. При нульових значеннях всіх координат вектора (асерційних) моніторів  $A^{rs} = 0$ , фіксується справний стан усього виробу. Якщо в розглянутій таблиці зафіксовані всі векторні хог-суми, не рівні нулю  $V_j^{rs} \oplus A^{rs} = 1$ , то корекції підлягає тест, побудований для перевірки даної функціональності.

Таким чином, представлена на рис 3.8 граф-схема дає можливість ефективно здійснювати сервісне обслуговування як завгодно складної технічної системи. Переваги такого движка, інваріантного до рівнів ієрархії, полягають у простоті підготовки й подання діагностичної інформації у вигляді мінімізованої таблиці активізації функціональних блоків або деструктивних компонентів (несправності, дефекти) на тестових сегментах.

### 3.7 Висновки

1. Представлено структурну модель відношень на множині із чотирьох основних компонентів технічної діагностики (функціональність, пристрій, тест, дефекти), що характеризується повним хог-взаємодією всіх вершин графа й транзитивна оборотність кожної тріади відношень, що дозволяє визначити й класифікувати шляхи розв'язок практичних завдань, включаючи синтез тестів, моделювання несправностей і пошук дефектів.

2. Запропоновано вдосконалені методи синтезу тестів для функціональностей, заданих матричними формами опису поведінки цифрових компонентів, які відрізняються паралелізмом векторних операцій над таблицями, що дає можливість істотно ( $\times 2$ ) підвищити швидкодію обчислювальних процедур.

3. Процес-моделі й методи синтезу тестів для функціональностей і діагностування порушень можуть бути використані як убудовані компоненти інфраструктури сервісного обслуговування цифрових систем на кристалах із застосуванням стандарту граничного сканування IEEE 1500.

5. Запропоновано вдосконалена процес-модель визначення функціональних порушень у програмному або апаратному виробі, що відрізняється використанням хог-операції, що дає можливість підвищити швидкодію діагностування одиночних або кратних дефектів (функціональних порушень) на основі паралельного аналізу таблиці ФН за допомогою логічних векторних операцій and, or, хог.

6. Розроблено структурну модель у вигляді мультидерева й метод (движок) його швидкого обходу, які відрізняються інваріантністю до рівнів ієрархії виробу для діагностування функціональних компонентів і несправностей на тестових сегментах, що дає можливість ефективно, у реальному масштабі часу, здійснювати сервісне обслуговування як завгодно складної технічної системи.

## 4 ІНФРАСТРУКТУРА ЛОГІЧНОГО АСОЦІАТИВНОГО МУЛЬТИПРОЦЕСОРА

Пропонується інфраструктура забезпечення паралельного аналізу табличних або матричних форм асоціативних відносин для пошуку, розпізнавання й прийняття рішень в  $n$ -мірному векторному логічному просторі на основі використання запропонованої архітектури мультипроцесора. Розглядаються векторно-логічні процеси-моделі актуальних прикладних завдань, у тому числі - убудоване діагностування й відновлення працездатності компонентів цифрових систем на кристалах, де якість розв'язок оцінюється уведеною неарифметичною метрикою взаємодії булевих векторів [101, 103, 106, 115-117, 119, 121].

### 4.1 Вступ

Ідея дослідження – вилучити з комп'ютера «великовагову» арифметику й трансформувати ресурси, що звільнилися, для створення інфраструктури векторно-логічних обчислень для швидкого пошуку, розпізнавання й прийняття рішень при аналізі інформаційного простору за допомогою примітивних операцій: and, or, not, xor. Спеціалізація комп'ютерного виробу, орієнтована на використання тільки логічних операцій, дає можливість істотно ( $\times 100$ ) підвищити швидкодія розв'язок неарифметичних завдань. Виключення арифметичних операцій, використання паралелізму алгебри векторної логіки, мультипроцесорність архітектури створюють ефективну інфраструктуру, що поєднує математичну й технологічну культуру для розв'язок прикладних завдань. Ринкова привабливість логічного асоціативного мультипроцесора (Logical Associative MultiProcessor – LAMP) визначається тисячами старих і нових логічних по своїй природі завдань, які в цей час вирішуються неефективно на надлишкових універсальних комп'ютерах з потужним

арифметичним процесором. От деякі актуальні для ІТ-ринку проблеми: 1. Аналіз, синтез і корекція синтаксичних і семантичних мовних конструкцій (реферування, виправлення помилок, оцінювання якості текстів). 2. Розпізнавання відео- і аудіо-образів на основі їхнього подання векторними моделями істотних параметрів у дискретному просторі. 3. Сервісне обслуговування складних технічних виробів і відновлення працездатності в процесі їхнього функціонування. 4. Тестування знань і експертне обслуговування об'єктів або суб'єктів для визначення їх валідності. 5. Ідентифікація об'єкта або процесу для ухвалення розв'язок в умовах невизначеності. 6. Точний пошук заданої вектором параметрів інформації в Internet. 7. Цілевказівка у винищувачі або в автоматичній системі посадки лайнера, що працює в реальному мікросекундному діапазоні часу. 8. Розведення об'єктів у часі й у просторі для диспетчерської служби аеропорту або оптимізація інфраструктури міського транспорту з метою виключення колізій. Практично всі згадані завдання вирішуються в реальному масштабі часу, є ізоморфними за логічною структурою процес-моделей, що використовують сукупність взаємозалежних асоціативних таблиць. Для їхнього розв'язок необхідна спеціалізована апаратна платформа (LAMP), орієнтована на паралельне виконання процедур пошуку, розпізнавання й прийняття рішень, оцінюваних шляхом використання неарифметичного критерію якості.

Мета дослідження – істотне підвищення швидкодії процедур пошуку, розпізнавання й прийняття рішень шляхом мультипроцесорної й паралельної реалізації асоціативно-логічних векторних операцій для аналізу графових і табличних структур даних у векторному логічному просторі без використання арифметичних операцій.

Для досягнення поставленої мети необхідно розробити: 1) Неарифметичну метрику оцінювання векторно-логічних рішень у кібернетичному просторі. 2) Структури даних і процес-моделі розв'язок

актуальних завдань. 3) Архітектуру логічного асоціативного мультипроцесора й показати шляхи його практичного використання.

Об'єктом дослідження є інфраструктура процесів пошуку, розпізнавання й прийняття рішень у векторно-логічному просторі на основі використання алгебри векторної логіки, обчислювальної архітектури аналізу асоціативно-логічних структур даних і неарифметичного інтегрального критерію якості.

У процесі досліджень використані джерела науково-технічної інформації: асоціативно-логічні структури даних для розв'язок інформаційних завдань [63, 69, 88-90]; апаратна платформа векторно-логічного аналізу інформації [62, 65, 67, 68, 91]; моделі й методи дискретного аналізу й синтезу об'єктів кіберпростору [65, 92-96]. Аналоги апаратної реалізації функціональностей представлені також у роботах [97-100]. Тут мова йде про реалізацію функціональності обробки зображення в режимі реального часу за допомогою So (MX-So) на основі мультипроцесора з архітектурою MX-So (три MX-Ядра, хост-процесор і введення / виводу, периферійні модулі). MX-Ядро, масовий паралелізм (1024), гнучкий SIMD-Процесор. Тактова частота –162 МГц.

#### 4.2 Скалярна метрика для оцінювання розв'язку

Дискретний векторно-логічний простір – кіберпростір – сукупність взаємодіючих по відповідній метриці інформаційних процесів і явищ, описуваних векторами (кортежами) логічні змінні й комп'ютерні системи, що використовують як носій, і мережі [107, 108].

Метрика – спосіб виміру відстані в просторі між компонентами процесів або явищ, описаних векторами логічних змінних. Відстань у кіберпросторі це - хог-відношення між парою векторів, що позначають компоненти процесу або явища, що відрізняє його від кодової відстані по Хемінгу. Відстань, похідна (булева), ступінь зміни, розходження або близькості є ізоморфні поняття, пов'язані з визначенням відносини двох компонентів процесу або явища.

Поняття близькості (відстані) компонентів у кіберпросторі є міра їхнього розходження. Похідна – міра бінарного відношення динамічних або статичних компонентів у процесах або явищах. Процедури порівняння, виміру, оцінювання, розпізнавання, тестування, діагностування, ідентифікування, є спосіб визначення відносини при наявності не менш чим двох об'єктів.

Компонент простору представлений  $k$ -мірним (двійковим) вектором  $a = (a_1, a_2, \dots, a_j, \dots, a_k)$ ,  $a_j = \{0,1\}$ , де кожна його координата визначена у двійковому алфавіті. Нуль-вектор є  $k$ -мірний кортеж, всі координати якого дорівнюють нулю:  $a_j = 0, j = \overline{1, k}$ .

Для розуміння наступного матеріалу необхідно ввести деякі допущення й визначення. Вхідний вектор  $m = (m_1, m_2, \dots, m_j, \dots, m_k)$ ,  $m_j \in \{0,1,x\}$  і аналізований об'єкт  $A = (A_1, A_2, \dots, A_j, \dots, A_k)$ ,  $A_j = \{0,1,x\}$ , що також представлений вектором, мають однакову розмірність  $k$ . Ступінь приналежності  $m$ -вектора до  $A$  позначається як  $\mu(m \in A)$ .

Визначення 4.1. Інтегральна теоретико-множинна метрика для оцінювання якості запиту є функція взаємодії багатозначних по координатах векторів  $m \Delta A$ , що визначається середньою сумою трьох нормованих параметрів: кодова відстань  $d(m, A)$ , функція приналежності  $\mu(m \in A)$  й функція приналежності  $\mu(A \in m)$ :

$$Q = \frac{1}{3}[d(m, A) + \mu(m \in A) + \mu(A \in m)],$$

$$d(m, A) = \frac{1}{n}[n - \text{card}[(i : m_i \cap A_i = \emptyset)_1^k]]; \mu(m \in A) = 2^{c-a}; \mu(A \in m) = 2^{c-b}; \quad (4.1)$$

$$a = \text{card}[(i : A_i = x)_1^k]; b = \text{card}[(i : m_i = x)_1^k]; c = \text{card}[(i : m_i \cap A_i = x)_1^k];$$

Пояснення. Перетинання (об'єднання) векторів – є векторна операція, заснована на відповідних координатних теоретико-множинних операціях.



Операції координатного перетинання й об'єднання (4.1) визначені в алфавіті Кантора  $A = \{0,1, x = \{0,1\}, \emptyset\}$ . Нормування параметрів дозволяє оцінити рівень взаємодії векторів у чисельному інтервалі  $[0,1]$ . Якщо зафіксовано граничне максимальне значення кожного параметра, рівне 1, то вектори рівні між собою. Мінімальна оцінка,  $Q = 0$ , фіксується у випадку повної розбіжності векторів по всім  $n$  координатах. Якщо потужність покоординатного перетинання  $m \cap A = m$  дорівнює половині потужності простору вектора  $A$ , то функції приналежності і якості рівні:

$$\mu(m \in A) = \frac{1}{2}; \mu(A \in m) = 1; d(m, A) = 1; Q(m, A) = \frac{5}{2 \times 3} = \frac{5}{6}.$$

Аналогічне значення буде мати параметр  $Q$ , якщо потужність перетинання  $m \cap A = A$  дорівнює половині потужності простору вектора  $m$ . Тут простір вектора є функція від числа координат, рівних  $x$ :  $q = 2^{\text{card } x}$ . Якщо потужність покоординатного перетинання  $\text{card}(m \cap A)$  дорівнює половині потужностей просторів векторів  $A$  і  $m$ , то функції приналежності мають значення:

$$\mu(m \in A) = \frac{1}{2}; \mu(A \in m) = \frac{1}{2}; d(m, A) = 1; Q(m, A) = \frac{4}{2 \times 3} = \frac{4}{6} = \frac{2}{3}.$$

Слід підкреслити, якщо перетинання двох векторів дорівнює порожній множині  $\exists i(m_i \cap A_i) = \emptyset$ , то ступінь двійки від символу «порожньо» дорівнює нулю:  $2^{\text{card}(m \cap A) = \emptyset} = 2^{\emptyset} = 0$ . Це дійсно означає, що кількість загальних крапок при перетинанні двох просторів, формованих двома векторами, дорівнює нулю. Векторно-логічний критерій якості розв'язок  $Q$  визначає

взаємодія компонентів (векторів)  $m$  і  $A$  при аналізі асоціативних структур даних:

$$\begin{aligned}
 Q &= d(m, A) + \mu(m \in A) + \mu(A \in m), \\
 d(m, A) &= \text{card}(\{i : m_i \oplus A_i = U, i = 1, \dots, k\}); \\
 \mu(m \in A) &= \text{card}(\{i : A_i = U, i = 1, \dots, k\}) - \text{card}(\{i : m_i \oplus A_i = U, i = 1, \dots, k\}); \\
 \mu(A \in m) &= \text{card}(\{i : m_i = U, i = 1, \dots, k\}) - \text{card}(\{i : m_i \oplus A_i = U, i = 1, \dots, k\}); \\
 U &= \begin{cases} 1 \leftarrow \{m_i, A_i\} = \{0, 1\}; \\ x \leftarrow \{m_i, A_i\} = \{0, 1, x\}. \end{cases}
 \end{aligned} \tag{4.2}$$

Представлені тут векторні логічні операції ізоморфні  $(\wedge, \vee, \oplus, \neg) \approx (\cap, \cup, \Delta, \sim)$  теоретико-множинними. Для того, щоб остаточно виключити арифметичні операції при підрахунку векторно-логічного критерію якості, необхідно логічно об'єднати три оцінки (4.2) в одну:

$$\begin{aligned}
 Q &= d(m, A) \vee \mu(m \in A) \vee \mu(A \in m) = m \oplus A \vee A \wedge \overline{m} \wedge \overline{A} \vee m \wedge \overline{m} \wedge \overline{A} = \\
 &= m \oplus A \vee [A \wedge (\overline{m} \vee \overline{A})] \vee [m \wedge (\overline{m} \vee \overline{A})] = m \oplus A \vee [A\overline{m} \vee A\overline{A} \vee m\overline{m} \vee m\overline{A}] = \\
 &= (A\overline{m} \vee m\overline{A}) \vee [A\overline{m} \vee A\overline{A} \vee m\overline{m} \vee m\overline{A}] = A\overline{m} \vee m\overline{A} \vee A\overline{m} \vee A\overline{A} \vee m\overline{m} \vee m\overline{A} = \\
 &= m \oplus A.
 \end{aligned}$$

Процедура обчислення векторного критерію залежить від значності алфавіту:

$$Q' = \begin{cases} m \oplus A \leftarrow \{m_i, A_i\} = \{0, 1\}; \\ m\Delta A \leftarrow \{m_i, A_i\} = \{0, 1, x\}. \end{cases} \tag{4.3}$$

Критерій якості  $Q$  однозначно визначає три форми взаємодії двох будь-яких об'єктів в  $n$ -мірному векторному логічному просторі: відстань або дві функції приналежності. При ненульовій відстані по Хемінгу функції приналежності втрачають зміст, оскільки простору двох векторів у цьому

випадку не перетинаються. У протилежному випадку – кодова відстань дорівнює нулю - взаємодія об'єктів оцінюється по функціях приналежності.

Для порівняння критеріїв якості необхідно визначати число одиниць у кожному векторі без виконання операцій підсумовування. Це можна зробити за допомогою регістра зрушення [62] (рис 4.1), що дозволяє за один такт виконати процедуру slc (shift left bit crowding) — зрушення вліво з одночасним ущільненням всіх одиничних координат n-розрядного двійкового вектора. Для наборів  $m = (110011001100)$ ,  $A = (000011110101)$  визначення якості їхньої взаємодії по формулах (4.5) представлено у вигляді наступних процедур (нульові координати відзначені крапками):

$m$	1 1 . . 1 1 . . 1 1 . .
$A$	. . . . 1 1 1 1 . 1 . 1
$m \wedge A$	. . . . 1 1 . . . 1 . .
$\overline{m \wedge A}$	1 1 1 1 . . 1 1 1 . 1 1
$d(m, A) = m \oplus A$	1 1 . . . . 1 1 1 . . 1
$\mu(A \in m) = m \wedge \overline{m \wedge A}$	1 1 . . . . . . 1 . . .
$\mu(m \in A) = A \wedge \overline{m \wedge A}$	. . . . . . 1 1 . . . 1
$Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$	1 1 . . . . 1 1 1 . . 1
$Q(m, A) = (6/12)$	1 1 1 1 1 1 . . . . . .

Тут сформована оцінка взаємодії векторів і, що саме головне, одиничні координати рядка  $Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$  ідентифікують всі істотні змінні, по яких взаємодія векторів не відповідає критерію якості. У скалярному критерії така структурна інформація назавжди втрачена. Як вибрати кращий розв'язок, якщо їх декілька?

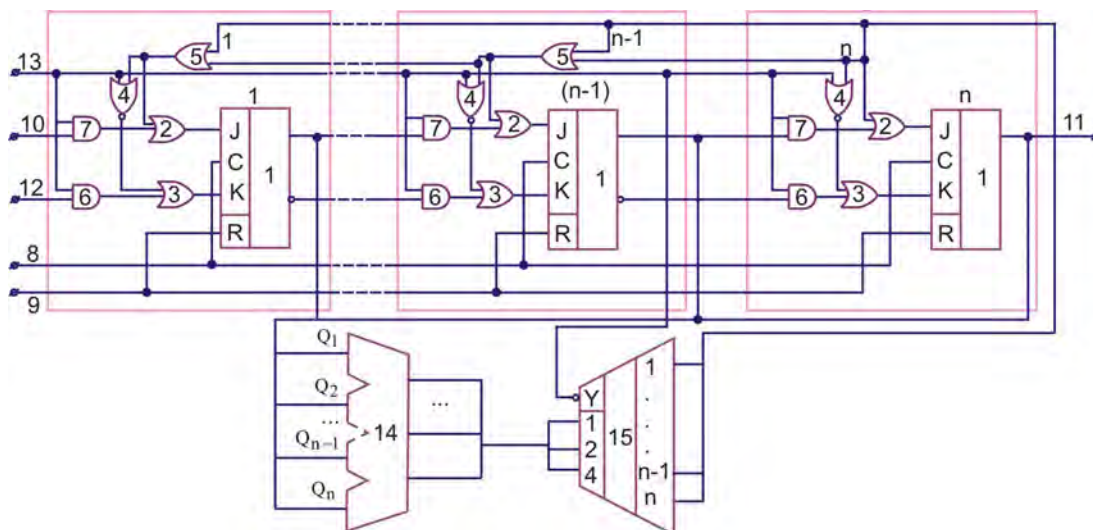


Рисунок 4.1 – Регістр зсуву й ущільнення одиниць

### 4.3 Архітектура логічного асоціативного мультипроцесора

На основі раніше уведеної метрики кіберпростору можна формувати квазіоптимальний інформаційний віртуальний простір діагностування  $S_0$ . Інформаційна структура Cyber Space Diagnosis повинна бути ієрархічної й замкнутої, як глобально, так і локально, на будь-якому рівні ієрархії. Елементарний осередок структури простору повинна бути трикутною. Це забезпечить істотне зменшення інформаційних обсягів Cyber Space, у межі - на третину. Це означає підвищення продуктивності всіх приймально-передавачів і пам'ятей на 33%? Дане твердження пов'язане з аксіомою транзитивного замикання кінцевого числа  $(1,2,3,4,\dots,n)$  крапок в інформаційному векторно-логічному просторі:

- 1)  $d_1 = 0$ ;
- 2)  $d_1 \oplus d_2 = 0$ ;
- 3)  $d_1 \oplus d_2 \oplus d_3 = 0$ ;
- 4)  $d_1 \oplus d_2 \oplus d_3 \oplus d_4 = 0$ ;
- 5)  $d_1 \oplus d_2 \oplus \dots \oplus d_i \oplus \dots \oplus d_n = 0$ .

Розглядаючи всі можливі варіанти транзитивного замикання з позиції мінімізації умов для відновлення інформації, очевидним представляється факт відновлення третьої сторони трикутника по двох відомим. У цьому випадку досить, наприклад, передати по каналі зв'язку дві сторони, щоб відновити третю. Скорочення переданих обсягів у цьому випадку дорівнює 33%. У всіх інших випадках зменшення обсягів інформації буде меншим. Наприклад, для замкнутого чотирикутника необхідно передати три відстані або сторони, щоб відновити четверту, виграш – 25%. Таким чином, площинна інтерпретація Cyber Space Diagnosis буде мати вигляд, представлений на рис 4.2.

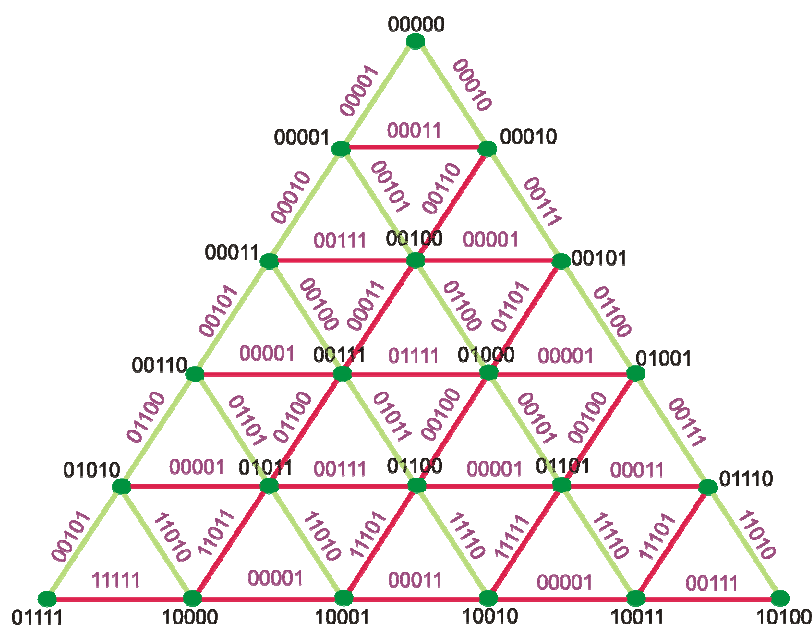


Рисунок 4.2 – Triangle Cyber Space

Для того щоб відновити всю згадану структуру, необхідно мати (передати) уже не 2/3 обсягу інформації, що відповідає кожним двом сторонам із трьох всіх трикутників, представлених зеленим (більше світлим) кольором. Для даної структури досить передати сторони, задані зеленим кольором, що становить 44% від обсягу кодів-відстаней. Інші 56% компонентів трикутників можуть бути визначені за допомогою аксіоми:  $d_1 \oplus d_2 \oplus d_3 = 0 \rightarrow d_3 = d_1 \oplus d_2$ . У загальному випадку функціональна

залежність відносини відновлюваних сторін замкнутого в трикутник простору до загального числа  $n$  шарів трикутної структури визначається вираженням:

$$\eta = \frac{\sum (i+1)}{\sum (3 \times i)} \Big|_{i=1, n} = \frac{1}{3} \left( \frac{2}{n+1} + 1 \right) = \frac{1}{3} \cdot \frac{2+n+1}{n+1} = \frac{1}{3} \cdot \frac{n+3}{n+1} = \frac{n+3}{3(n+1)}.$$

Графік відношення кількості переданої інформації до її повного відновлюваного обсягу відстаней у трикутному кібернетичному просторі представлений на рис 4.3. Граничне значення графіка прагне до 33%. Це означає – для більших інформаційних обсягів трикутного простору досить передати третю частину всіх кодів-відстаней, щоб потім відновити повний простір кодів.

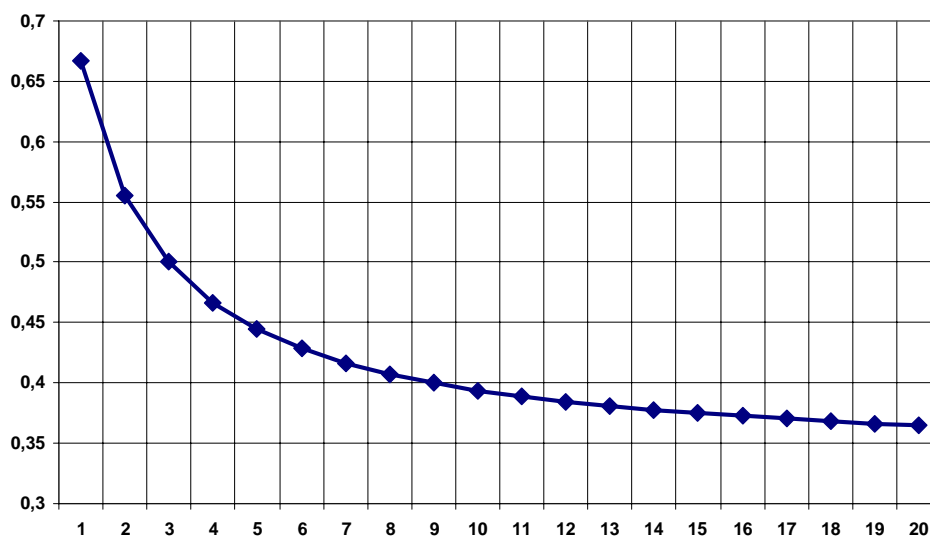


Рисунок 4.3 – Функція числа відстаней для відновлення простору

Трикутна метрика простору є сама економічна, оскільки вона створює самі короткі відстані й шляхи між об'єктами, завдяки наявності транзитивного замикання. Сусідство об'єкта із шістьма крапками робить метрику більше кращою у порівнянні з «Манхетеном» у частині вибору шляхів для досягнення

об'єкта в просторі. Оптимальна структура мультипроцесорної системи повинна бути складена із трикутників. Тут оптимальність визначається співвідношенням довжини шляхи між будь-якими двома крапками й загальним числом сторін, що формують трикутний простір. Довжина шляху тут завжди буде не гірше, ніж в «Манхетені», але загальне число зв'язків більше на одну діагональ у кожному чотирикутнику. Це дає можливість існування прямих контактів для 6 сусідів, що істотно для мультипроцесора. «Манхетен» з діагональними зв'язками має ту ж довжину шляху, що й трикутний простір, але маючи при цьому на один зв'язок більше. Тут задіяна друга діагональ чотирикутника, що забезпечує сусідство з вісьма вершинами простору. Оскільки трикутник сама примітивна фігура, що формує площина, то вага однієї сторони, що не слід зберігати для ідентифікації простору, буде максимальним у трикутнику. Далі зібрані в систему трикутники дають уже 66% структурних компонентів, які можна не описувати при формуванні простору, але які при бажанні можна відновити. Всі інші примітивні площинні фігури дають менший вигравш при формуванні ними Cyber Space.

Для аналізу більших інформаційних обсягів логічних даних існує трохи практично орієнтованих технологій: 1. Використання робочої станції для послідовного програмування завдання, де вартість її розв'язок, а також тимчасові витрати дуже високі. 2. Розробка спеціалізованого паралельного процесора на основі PLD. Високий паралелізм обробки інформації компенсує порівняно низьку в порівнянні з CPU тактову частоту. Такий схемотехнічний розв'язок з можливістю перепрограмування є по продуктивності вигравшним варіантом. Недолік - відсутність гнучкості, характерної програмним методам розв'язок логічних завдань і висока вартість реалізації системи на кристалі PLD при більших обсягах промислового випуску виробу. 3. Краще розв'язок може бути отримане в результаті об'єднання достоїнств центрального процесорного пристрою, програмувальної логічної матриці й замовлений СБИС, таких як гнучкість програмування, можливість коректування вихідних кодів; мінімальне

число команд і прості схемотехнічні розв'язок апаратної реалізації мультипроцесора; розпаралелювання логічних процедур на структурі однобітових процесорів. Реалізація обчислювача в кристалі замовлений СБИС дає можливість одержувати максимальну тактову частоту, мінімальну вартість мікросхеми при більших обсягах випуску виробу, низьке енергоспоживання. Об'єднання переваг перерахованих технологій визначає базову конфігурацію обчислювача, що має структуру (рис 4.4), що складається з 16-ти векторних секвенсорів - пристроїв послідовного керування (УПУ), кожний з яких, включаючи граничні елементи, з'єднаний з вісьма сусідніми. Прототипом даного обчислювача є процесор PRUS, описаний у роботі [91]. Вісім з'єднань секвенсора дають можливість організувати ефективні шляхи прийому передачі даних, як по діагоналях, так і по ортогональних зв'язках між процесорами. Така структура є надлишковою стосовно трикутної системи взаємодії процесорів, але це дає можливість більш гнучко (по горизонталях, вертикалям і діагоналям) планувати обчислювальні процеси для обробки даних у паралельному режимі.

Занесення інформації в процесор виконується за класичною схемою процесу проектування, за винятком стадії розміщення й трасування, що замінюється фазою розподілу програм і даних між всіма логічними біт-процесорами, що працюють паралельно. Компілятор забезпечує розподіл даних по процесорах, задає час формування розв'язок на виході кожного з них, а також планує передачу отриманих результатів іншому процесору.

Логічний асоціативний мультипроцесор (ЛАМП) – це ефективна мережа процесорів, що обробляє дані й забезпечує обмін інформацією між компонентами мережі в процесі їхнього розв'язок. Проста схемотехніка кожного процесора дозволяє ефективно обробляти надвеликі масиви, що нараховують мільйони біт інформації, затрачаючи на це в сотні разів менше часу в порівнянні з універсальним процесором.



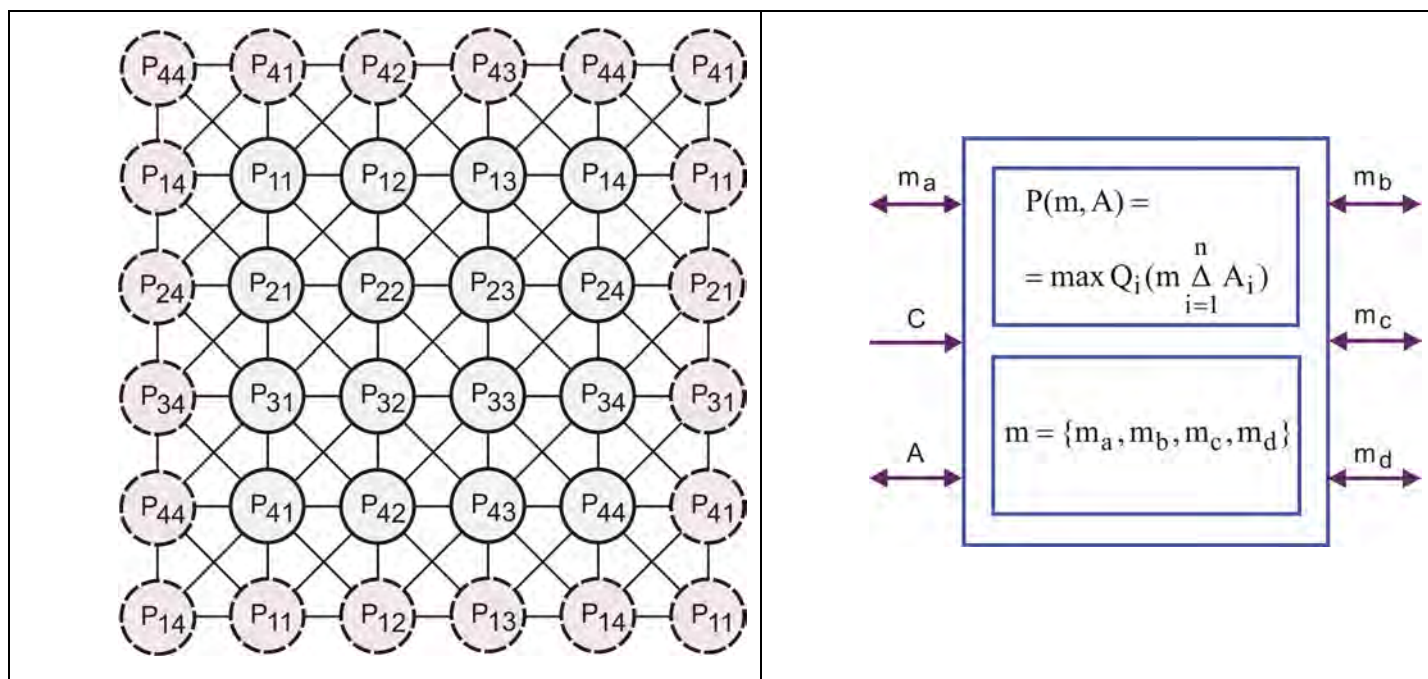


Рисунок 4.4 – Макроархітектура мультипроцесора й інтерфейс

Базова комірка – векторний процесор для обчислювача може бути синтезований на 200-х вентилях, що дає можливість мережа, що містить 4096 обчислювачів, легко реалізувати в кристалі замовлений СБИС, використовуючи сучасну кремнієву технологію. Оскільки витрати пам'яті для зберігання даних досить незначні, обчислювач може бути використаний при проектуванні систем керування в таких областях людської діяльності, як промислове виробництво, захист інформації, медицина, штучний інтелект, космонавтика, геологія, метеорологія. Особливий інтерес обчислювач представляє для цифрової обробки даних, розпізнавання образів і криптоанализу. Однак основне призначення ЛАМП - одержання квазіоптимального розв'язок в завданні пошуку й (або) розпізнавання з використанням компонентів архітектури, орієнтованих на виконання векторних логічних операцій:

$$P(m, A) = \min_{i=1}^n Q_i(m, A_i), m = \{m_a, m_b, m_c, m_d\}. \quad (4.4)$$

Інтерфейс системи, що відповідає даному функціоналові (4.4), представлений на рис 4.4. Усі компоненти  $\{A, m_a, m_b, m_c, m_d\}$  можуть бути як вхідними, так і вихідними. Двонаправлена деталізація інтерфейсу пов'язана з інваріантністю відносини всіх змінних, векторів,  $A$ -матриці й компонентів до входів і (або) виходам архітектури. Тому структурна модель системи обчислювача може бути використана для розв'язок будь-яких завдань прямій і зворотної імплікації в дискретному логічному просторі, у чому й полягає її відмінність від концепції автоматної моделі обчислювального пристрою з вираженими входами й виходами. Компоненти або регістри  $m = (m_a, m_b, m_c, m_d)$  використовуються для одержання розв'язок у вигляді буферних, вхідних і вихідних векторів, а також для ідентифікації оцінки якості задоволення вхідного запиту.

Один з можливих варіантів архітектури ЛАМПІ представлений на рис. 4.5. Основним компонентом є матриця  $P = [P_{ij}]$ ,  $\text{card}(4 \times 4)$ , що містить 16 вектор-процесорів, кожний з яких призначений для виконання п'яти логічних векторних операцій над пам'яттю даних, представленої у вигляді таблиці, розмірністю  $A = \text{card}(m \times n)$ .



Рисунок 4.5 – Архітектура ЛАМПІ і структура УПУ

У блоці інтерфейсу відбувається обмін даними й завантаження програми обробки даних у відповідну пам'ять команд. Блок керування ініціює виконання команд логічної обробки даних і синхронізує функціонування всіх компонентів

мультипроцесора. Блок ІР [67] призначений для сервісного обслуговування всіх модулів, діагностування дефектів і відновлення працездатності компонентів і пристрою в цілому. Елементарний логічний асоціативний процесор або УПУ (див. рис 4.5), що входить до складу обчислювача, містить логічний процесор LP, асоціативну (пам'ять) А-Матрицю для паралельного виконання базових операцій, блок векторів  $m$ , призначений для паралельного обслуговування рядків і стовпців А-Матриці, а також обміну даними в процесі обчислень, пам'ять прямого доступу СМ, що зберігає команди програми обробки інформації, автомат керування СУ виконанням логічних операцій, інтерфейс І зв'язку УПУ с іншими елементами й пристроями ЛАМП.

Логічний процесор (LP) (рис 4.6) виконує п'ять операцій (and, or not, xor, slc), що є базовими для створення алгоритмів і процедур інформаційного пошуку й оцінювання розв'язок. Модуль LP має мультиплексор, комутуючий один з п'яти операндів з обраним логічним векторним оператором. Сформований результат через мультиплексор (елемент or) заноситься в один із чотирьох операндів, обраний відповідною адресою.

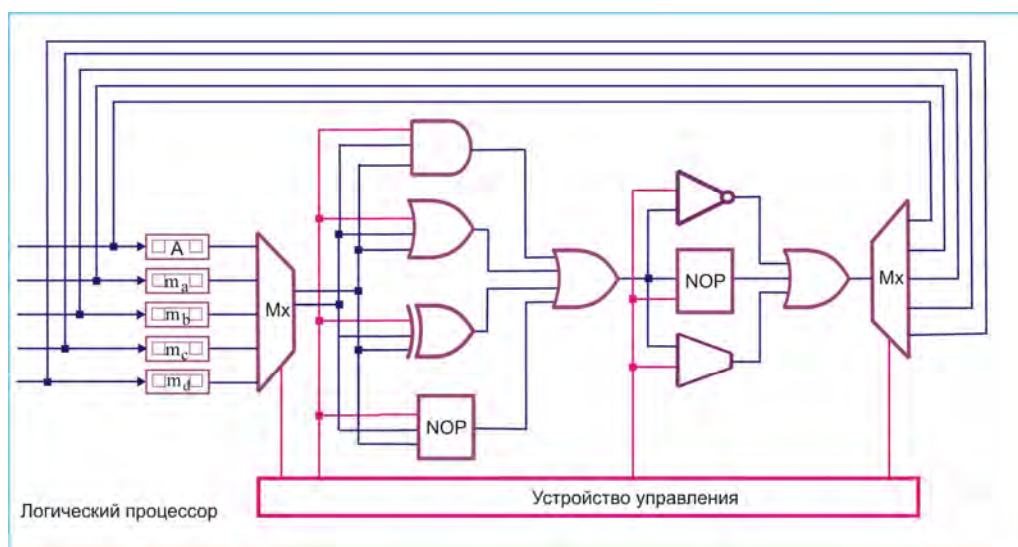


Рисунок 4.6 – Структура блоку логічних обчислень

Особливості реалізації логічного процесора полягають у наявності трьох бінарних (and, or, xor) і двох унарних (not, slc) операцій. Останні можна приєднувати до такту обробки регістрових даних, вибравши одну із трьох операцій (not, slc, nor - немає операції). Для підвищення ефективності роботи логічного пристрою вводяться два елементи з порожньою операцією. Якщо необхідно виконати тільки унарну операцію, то на рівні бінарних команд варто вибрати nor, що практично означає передачу даних через повторювач до другого рівня унарних операцій. Всі операції в LP – регістрові або регістрово-матричні. Останні призначені для аналізу вектор-рядків таблиці при використанні вхідного m-вектора як запиту для точного пошуку інформації. У блоці логічних обчислень припустиме наступне сполучення операцій і операндів:

$$C = \begin{cases} \{m_a, m_b, m_c, m_d\} \Delta A_i; \\ \{m_a, m_b, m_c, m_d\} \Delta \{m_a, m_b, m_c, m_d\}; \\ \{\text{not, nor, slc}\} \{m_a, m_b, m_c, m_d, A_i\}. \end{cases}$$

$$\Delta = \{\text{and, or, xor}\}.$$

Реалізація всіх векторних операцій блоку логічних обчислень, виконуваних з тактовою частотою 100 МГц, для одного УПУ в середовищі Verilog з наступної післясинтезною реалізацією в кристалі програмувальної логіки Virtex 4, Xilinx містить 2400 еквівалентних вентилів.

#### 4.4 Інфраструктура векторно-логічного аналізу

Інфраструктура – сукупність моделей, методів і засобів опису, аналізу й синтезу структур даних для розв’язок функціональних завдань. Модель (системна) – сукупність взаємозалежних, певних у просторі й часі компонентів, яка із заданою адекватністю описує процес або явище й використовується для досягнення поставленої мети при наявності обмежень і метрики оцінювання

якості розв’язок. Тут обмеження є апаратурні витрати, час розробки й виробництва до появи виробу на ринку ( time-to-market), підмети мінімізації. Метрика оцінювання розв’язок при використанні моделі визначена двійковим логічним вектором у дискретному булевому просторі. Концептуальна модель обчислювального виробу представлена сукупністю керуючих і операційного автоматів. У моделі функціональності використана ієрархічна технологія створення цифрових систем з локальною синхронізацією окремих модулів і одночасно глобальної асинхронністю функціонування всього пристрою [94].

Для деталізації структури векторного процесора й УПУ далі розглянутий аналітичні й структурні процес-моделі, що виконують аналіз А-Матриці по стовпцях або рядкам. Перша з них представлена на рис 4.7 і призначена для визначення множині припустимих рішень щодо вхідного запиту  $m_b$ , друга (рис 4.8) здійснює пошук оптимального розв’язок на множині рядків, знайдених за допомогою першої моделі в результаті їхнього аналізу.

$$m_{ai}^m = \bar{\vee} [(m_b \wedge A_i) \oplus m_b];$$

$$A_i = (m_b \wedge A_i).$$

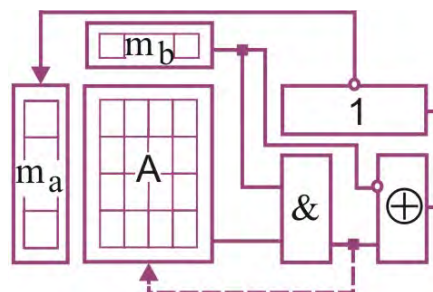


Рисунок 4.7 – Пошук всіх припустимих рішень

Можливо й самостійне застосування другої моделі, орієнтоване на визначення однозначного й багатозначного розв’язок при пошуку дефектів у цифровій системі.

Всі операції, виконувані двома процес-моделями – векторні. Модель аналізу рядків (див. рис 4.7) формує вектор  $m_a$  ідентифікації припустимих

$m_{ai} = 1$  або суперечливих  $m_{ai} = 0$  рішень щодо вхідної умови  $m_b$  за  $n$  тактів обробки всіх  $m$ -розрядних векторів таблиці  $A = \text{card}(m \times n)$ . Якість (допустимість) розв'язок визначається для кожної взаємодії вхідного вектора  $m_b$  й рядка  $A_i \in A$  на блоці (редукції) диз'юнкції. Матриця  $A$  може бути модифікована її перетинанням із вхідним вектором на основі використання операції  $A_i = (m_b \wedge A_i)$ , якщо необхідно виключити з  $A$ -Таблиці всі незначущі для розв'язок координати й вектори, відзначені одиничними значеннями вектора  $m_a$ .

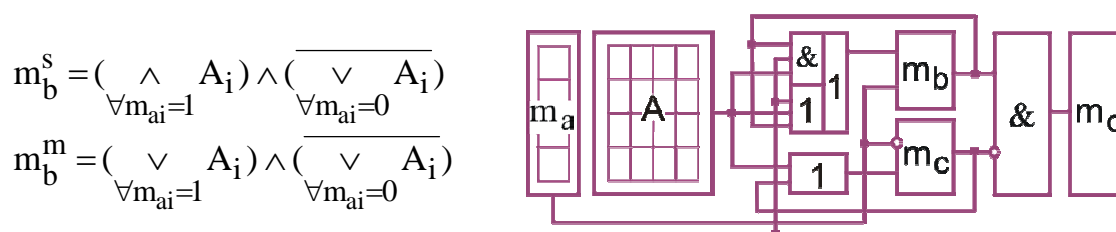


Рисунок 4.8 – Вибір оптимального розв'язок

Розв'язок завдань діагностування за допомогою аналізу рядків таблиці (див. рис 4.8) здійснюється так. Після виконання діагностичного експерименту формується двійковий вектор експериментальної перевірки  $m_a$ , що маскує  $A$ -Таблицю несправностей для пошуку одиночних або кратних дефектів. Вектори  $m_b$  й  $m_c$  використовуються для нагромадження результатів виконання операцій кон'юнкції й диз'юнкції. Потім виконується логічне вирахування (xor-операція) з першого регістра  $m_b$  вмісту другого вектора  $m_c$  з наступним записом результату в регістр  $m_d$ . Для реалізації другого рівняння, що формує множинне розв'язок, елемент and замінюється функцією or. У схемі використовується також змінна вибору режиму пошуку розв'язку: single або multiple. Як вхідна умова в моделі використаний вектор  $m_a$ , керуючий вибором

векторної операції  $\text{and}$ ,  $\text{or}$  для обробки одиничних  $A_i(m_{ai} = 1) \in A$  або нульових  $A_i(m_{ai} = 0) \in A$  рядків  $A$ -таблиці. У результаті виконання  $n$  тактів здійснюється нагромадження одиничних і нульових щодо значень координат вектора  $m_a$  рішень у регістрах  $A_1, A_0$ . Априорі в зазначені регістри заноситься вектор одиниць і нулів:  $A_1 = 1, A_0 = 0$ . Після обробки всіх  $n$  рядків  $A$ -Таблиці за  $n$  тактів виконується векторна кон'юнкція вмісту регістра  $A_1$  з інверсією регістра  $A_0$ , що формує результат у вигляді вектора  $m_b$ , де одиничні значення координат визначають розв'язок. У таблиці несправностей цифрового виробу одиничним координатам вектора  $m_b$  відповідають стовпці, що ототожнюються з номерами дефектів або несправних блоків, що підлягають відновленню або ремонту.

При сервісному обслуговуванні функціональних модулів можна на універсальній структурі системи векторного логічного аналізу вирішити оптимізаційне завдання відновлення працездатності. За допомогою мінімального числа ремонтних запасних рядків і (або) стовпців, наприклад пам'яті, необхідно забезпечити квазіоптимальне покриття всіх виявлених в осередках несправностей. Технологічна й математична складові векторної логіки в цьому випадку обумовлює простої схемотехнічний розв'язок для одержання квазіоптимального покриття (рис 4.9), переваги якого полягає в наступному: 1. Обчислювальна складність процедури: число векторних операцій, рівне числу рядків таблиці,  $Z = n$ . 2. Мінімум апаратних витрат: таблиця й два вектори  $(m_b, m_a)$  для зберігання проміжних покриттів і нагромадження результату у вигляді одиничних координат, що відповідають рядкам таблиці, які становлять квазіоптимальне покриття. 3. Відсутність класичного розподілу завдання покриття на пошук ядра покриття й доповнення. 4. Відсутність складних процедур маніпулювання осередками рядків і стовпців. Одержання не завжди оптимального покриття – недолік, що компенсується технологічністю векторної процедури, представленої на рис 4.9.



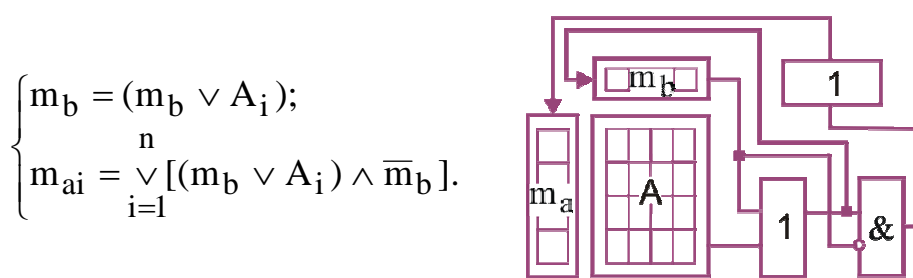


Рисунок 4.9 – Процес-модель пошуку квазіоптимального покриття

Операція редукції, що на останньому етапі перетворює векторний результат у біт  $m_{ai}$  вектора  $m_a$  по функції  $og\ m_{ai} = \bigvee [(m_b \vee A_i) \wedge \bar{m}_b]$ . У загальному випадку операція редукції в алгебрі векторних операцій записується у вигляді <бінарна операція><вектор>:  $\vee A_i, \wedge m, \bar{\wedge} (m \vee A_i)$ . Зворотна процедура – векторизація є конкатенація булевих змінних:  $m_a (a, b, c, d, e, f, g, h)$ . У процедурі пошуку покриття апріорі вектори  $m_b = 0$ ,  $m_a = 0$  стають рівними. Квазіоптимальне покриття накопичується за  $n$  тактів у векторі  $m_a$  послідовним зрушенням. Біти, що заносяться в реєстр  $m_a$ , формуються схемою  $og$ , що виконує редукцію після аналізу отриманого результату  $[(m_b \vee A_i) \wedge \bar{m}_b]$  на наявність одиниць.

Становить інтерес функціональна закінченість циклу діагностування, коли після одержання квазіоптимального покриття дана інформація використовується для відновлення працездатності дефектних комірок пам'яті [67]. Розмірність модуля пам'яті (13x15 осередків) не впливає на обчислювальну складність одержання покриття десяти дефектних осередків за допомогою резервних рядків і стовпців (рис. 4.10).

Для розв'язок оптимізаційної задачі виконується побудова таблиці покриття (див. рис. 4.10) несправних осередків, у якій рядка – резервні ресурси для покриття дефектів ( $C_2, C_3, C_5, C_7, C_8, C_2, R_2, R_4, R_5, R_7, R_8, R_9$ ), а стовпці – дефекти осередків ( $F_{2,2}, F_{2,5}, F_{2,8}, F_{4,3}, F_{5,5}, F_{5,8}, F_{7,2}, F_{8,5}, F_{9,3}, F_{9,7}$ ), що підлягають ремонту.



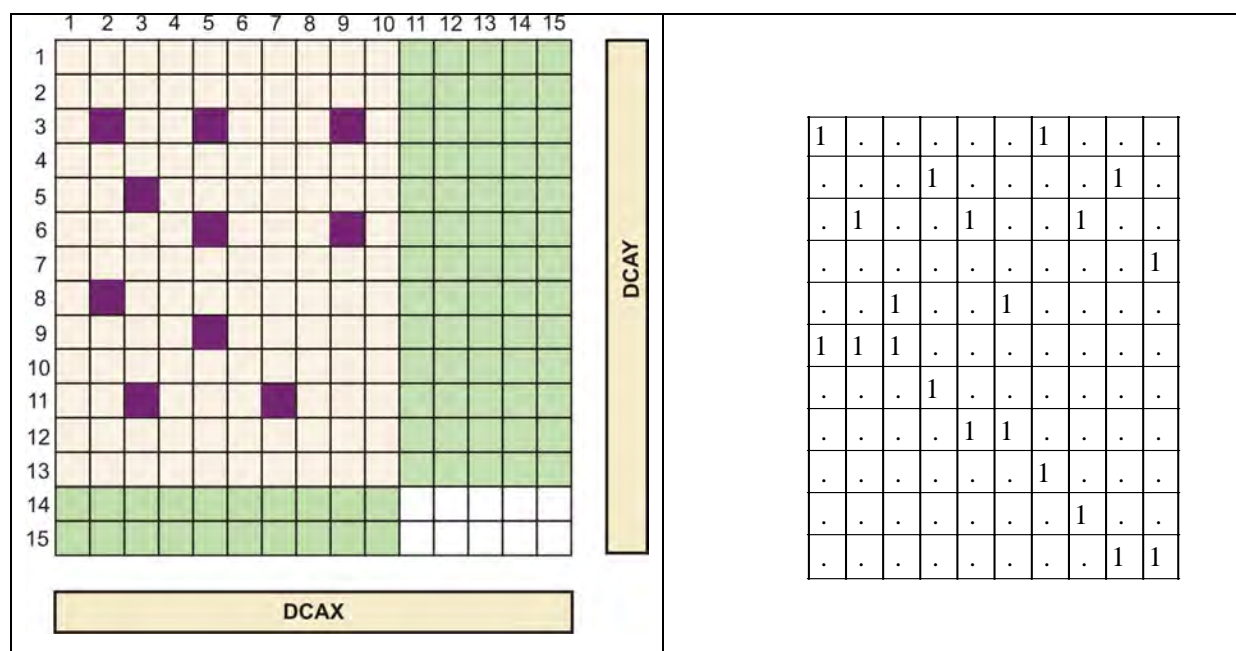


Рисунок 4.10 – Модуль пам'яті з резервом і таблиця покриття

При цьому стовпці відповідають координатам дефектних осередків, а рядка ідентифікують резервні компоненти (рядки й стовпці), які можуть відновити працездатність несправних координат. Модель обчислювального процесу, представлена на рис 4.8, дає можливість одержати оптимальне розв'язок у вигляді  $m_a = [1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$ , якому відповідає покриття  $R = \{C_2, C_3, C_5, C_7, C_8\}$ , як одне із трьох можливих мінімальних рішень  $R = C_2, C_3, C_5, C_7, C_8 \vee C_2, C_3, C_5, C_8, R_9 \vee C_2, C_5, C_8, R_4, R_9$  для таблиці несправностей. Технологічна модель убудованого діагностування й ремонту пам'яті (рис 4.11) має чотири компоненти: 1. Тестування модуля (Unit Under Test (UUT)) з використанням еталонної моделі (Model Under Test (MUT)) для формування вектора експериментальної перевірки  $m_a$ , розмірність якого відповідає числу тестових наборів. 2. Пошук дефектів на основі аналізу таблиці несправностей A. 3. Оптимізація покриття дефектних осередків ремонтними рядками й стовпцями на основі аналізу таблиці A. 4. Відновлення працездатності пам'яті за допомогою заміни адрес (Address Decoder (AD)) несправних рядків і

стовпців, представлених вектором  $m_a$ , на адреси компонентів з ремонтного запасу (Spare Memory (SM)) [67].

Процес-модель убудованого сервісного обслуговування працює в реальному масштабі часу й дозволяє підтримувати в працездатному стані, без втручання людини, цифрову систему на кристалі, що є доцільним розв'язком у випадку застосування технологій, пов'язаних з дистанційною експлуатацією виробу. Запропоновані процес-моделі аналізу асоціативних таблиць, а також уведені критерії якості логічних рішень дозволяють вирішувати завдання квазіоптимального покриття, діагностування дефектів програмних і (або) апаратних блоків. Модель векторних обчислень стала основою для розробки спеціалізованої мультипроцесорної архітектури, орієнтованої на пошук, розпізнавання й прийняття рішень про використання структур асоціативних таблиць.

Аналітична оцінка ефективності проектного розв'язок, спрямованого на виконання умов спеціалізації  $S_p$  і стандартизації  $S_t$  (рис 4.12) визначається мінімумом середнього значення наступних трьох взаємно суперечливих відносних і безрозмірних параметрів: рівень помилок проекту  $L$ , час верифікації й (або) тестування  $T$ , програмно-апаратна надмірність, обумовлена механізмами асерцій і (або) граничного сканування  $H$ .

Параметр  $L = 1 - Y^{(1-k)} = 1 - (1 - P)^{n(1-k)}$ , як доповнення до  $Y = (1 - P)^n$ , що характеризує вихід придатної продукції, залежить від тестопридатності проекту  $k$ , імовірності  $P$  існування несправних компонентів і числа невиявлених помилок  $n$ . Час верифікації визначається тестопридатністю проекту  $k$ , помноженої на структурну складність апаратно-програмної функціональності, віднесеної до загальної складності проекту в рядках коду або еквівалентних вентилів. Програмно-апаратна надмірність перебуває у функціональній залежності від складності асерційного коду або механізму граничного сканування, віднесеної до загальної складності проекту. При цьому асерційна, або скануюча, надмірність повинна забезпечувати задану глибину

діагностування помилок функціональності за час виходу виробу на ринок, визначене замовником.

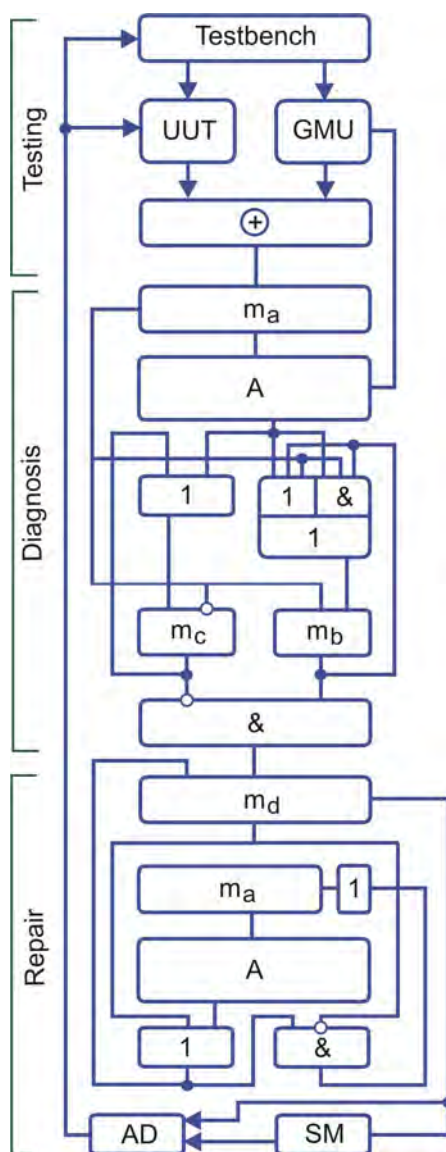


Рисунок 4.11 – Модель убудованого тестування й ремонту компонентів

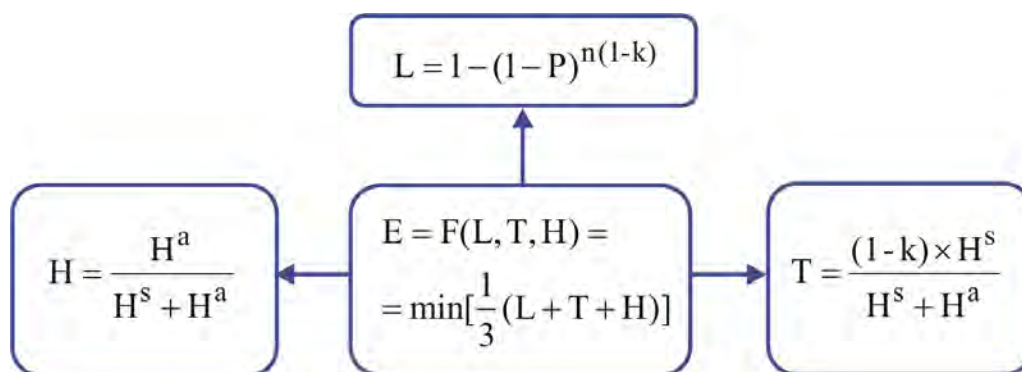


Рисунок 4.12 – Ефективність проектного розв’язок

#### 4.5 Апаратна імплементація компонентів мультипроцесора

У відповідності до структури процесора, запропонованої в підрозд. 4.4 (рис 4.4–4.6), була здійснена імплементація окремих компонентів в FPGA Virtex 6 шляхом використання мов опису апаратури VHDL, Verilog, системи проектування Xilinx ISE Design Suite 13, засобу моделювання й верифікації Xilinx ISIM. Результати синтезу й моделювання, а також трасування й розміщення компонентів мультипроцесора представлені на рис. 4.13 – 4.19. Результат синтезу модуля діагностування на основі використання векторних операцій, представлений на рис 4.11, показаний на рис.4.14. Процес моделювання верифікації базових змінних модуля діагностування з рис. 4.11 представлений на рис. 4.15. Фрагмент схемотехнічного розв’язок векторних логічних операцій у вигляді RTL-Моделі наведений на рис. 4.16. Процес моделювання й верифікації апаратної реалізації векторної логічної операції И представлений на рис. 4.17. Фрагмент схемотехнічного розв’язок для реалізації векторної операції зрушення показаний на рис. 4.18. Моделювання стану регістрів у процесі обміну інформацією між пам'яттю й логічними секвенсорами представлено на рис. 4.18.

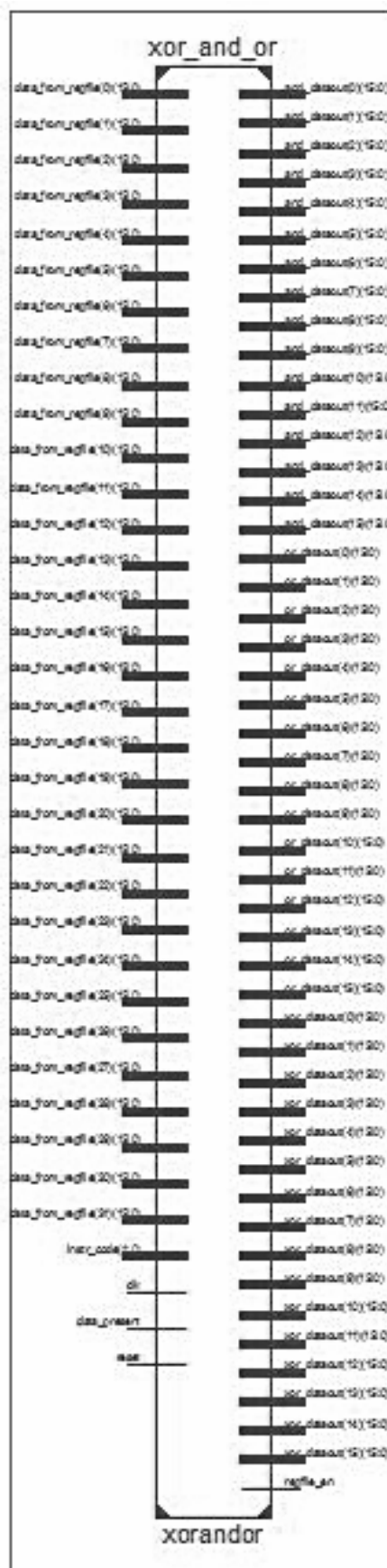


Рисунок 4.13 – Інтерфейс мультипроцесорної секції

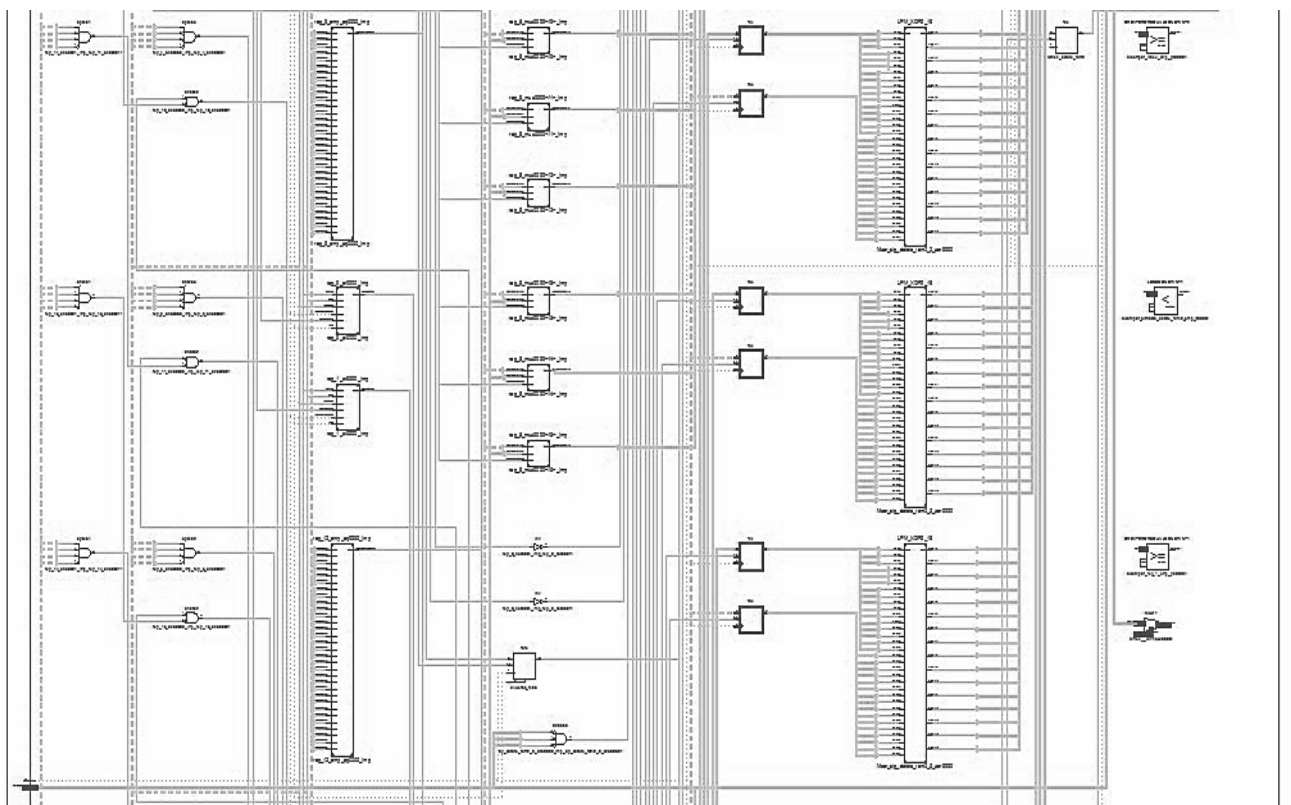


Рисунок 4.14 – Результат синтезу векторної операції Xor

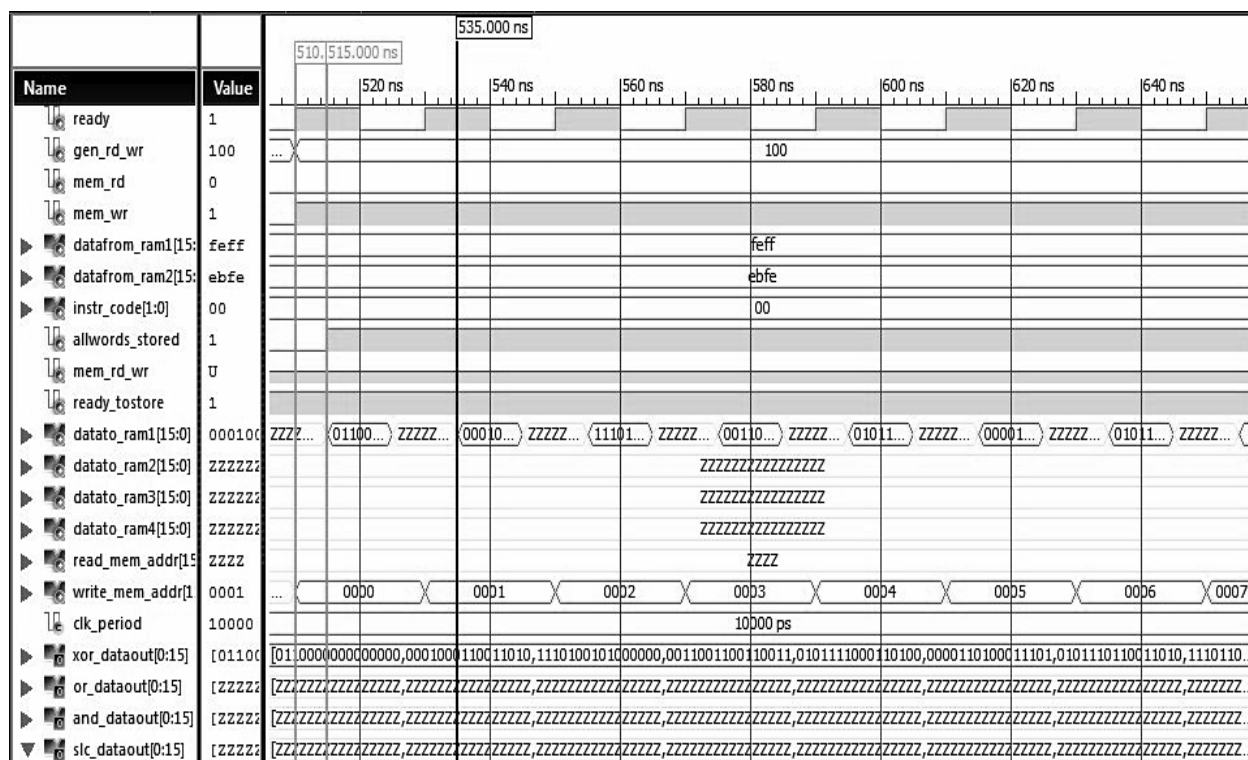


Рисунок 4.15 – Взаємодія векторів матриць у процесі діагностування

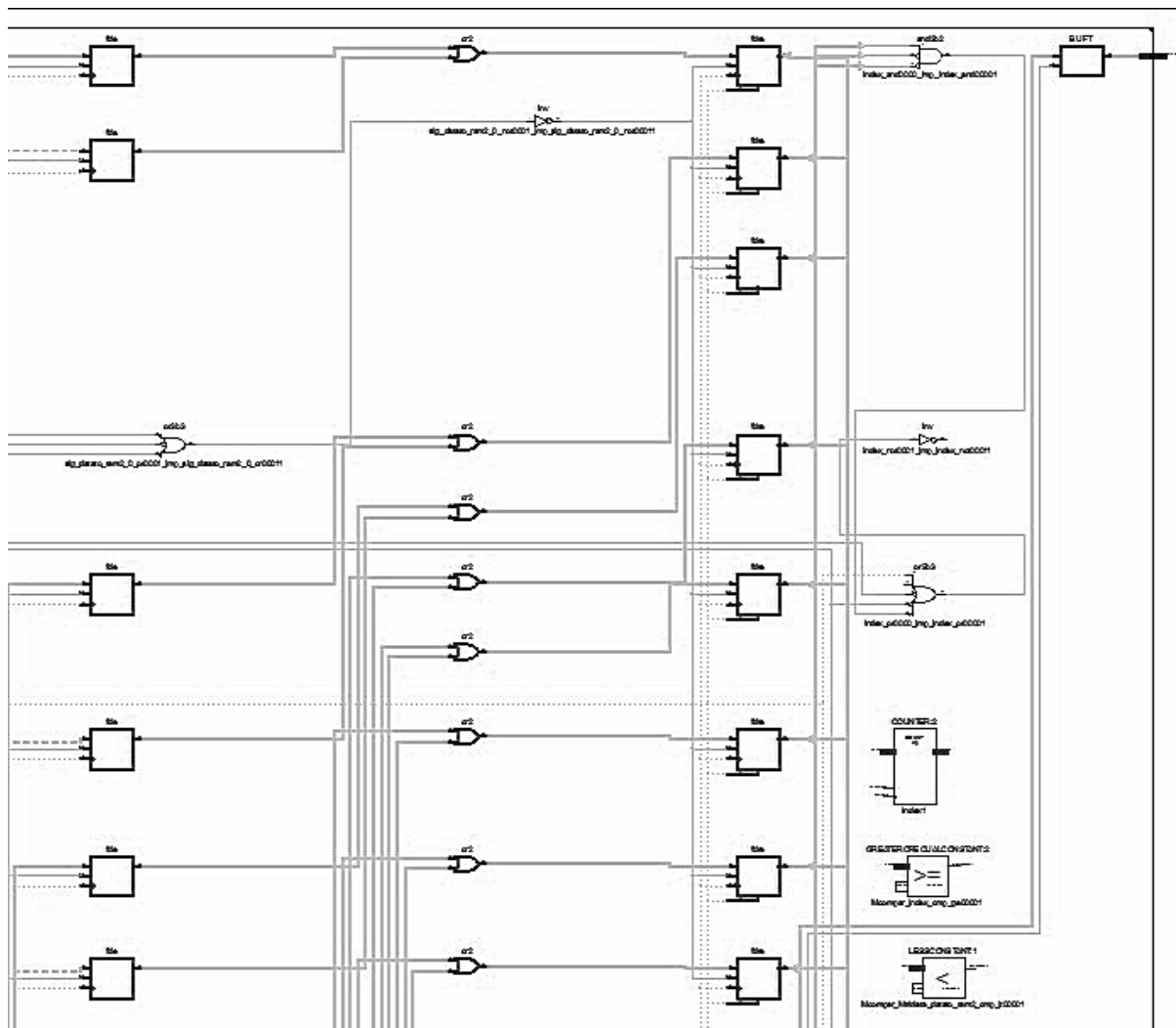


Рисунок 4.16 – Схемотехническое розв’язок векторної операції АБО

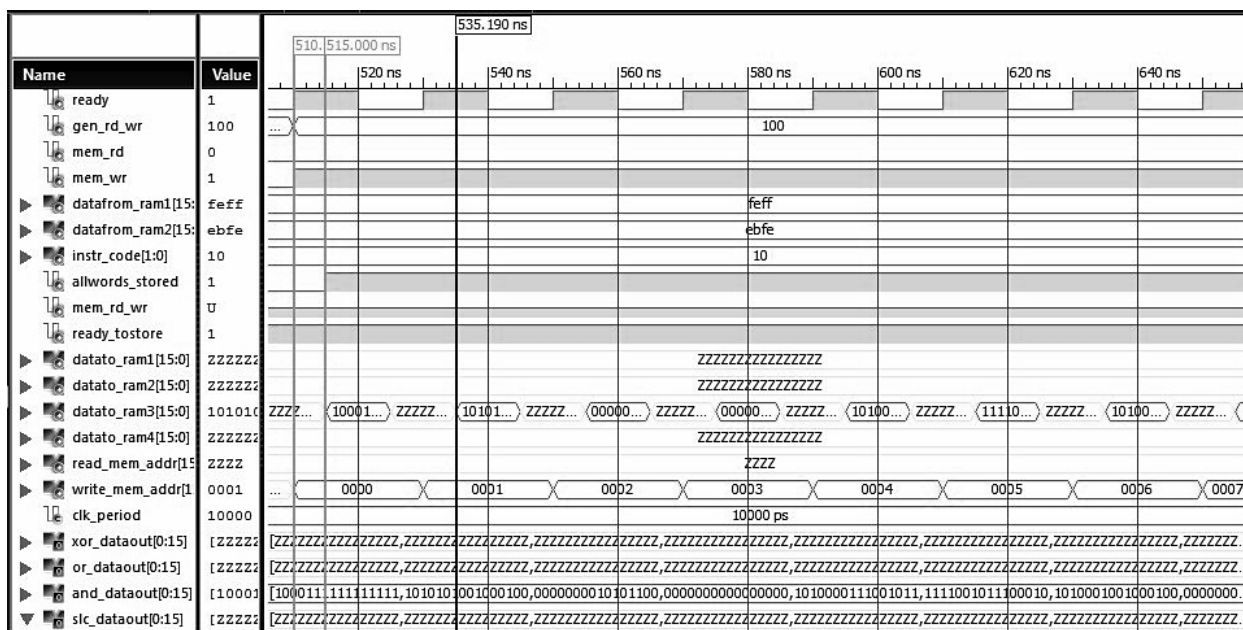


Рисунок 4.17 – Верифікація векторної операції И

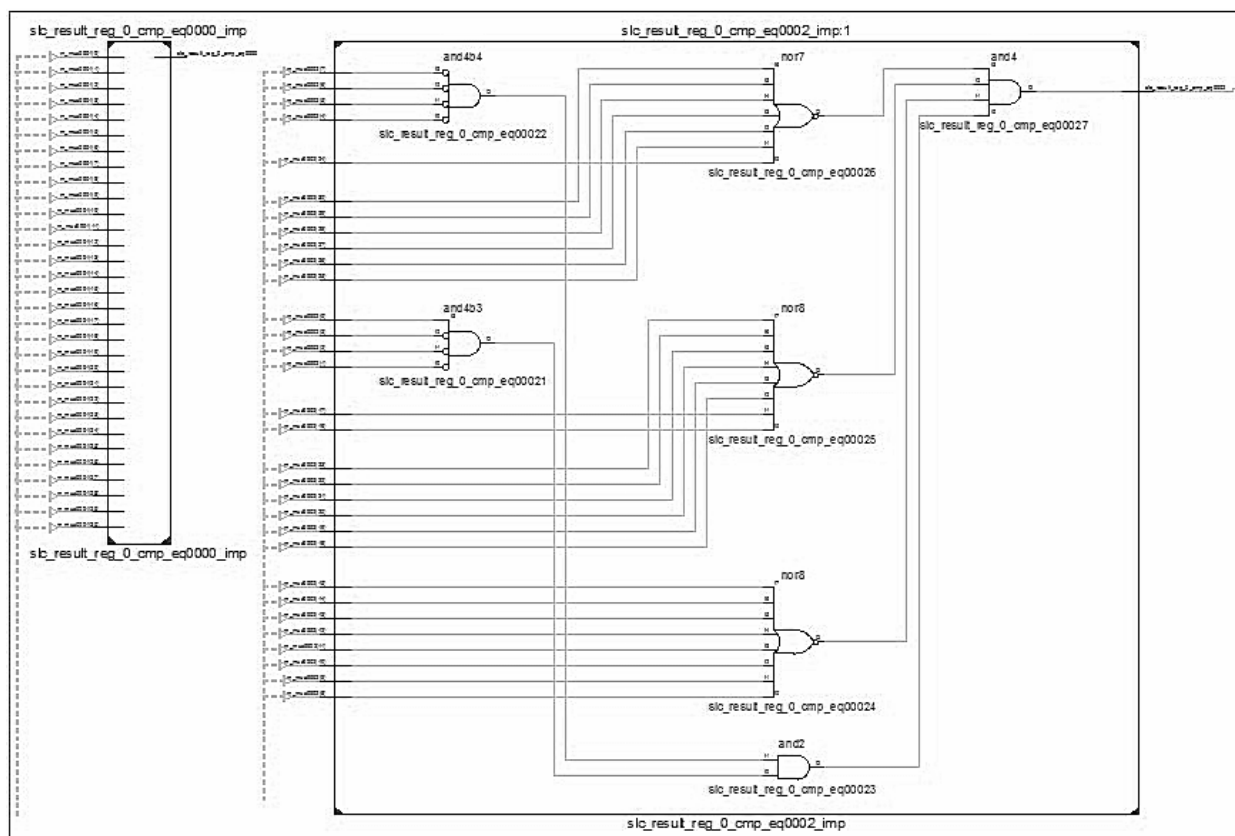


Рисунок 4.18 – Схемотехнічне розв'язок векторної операції SLC

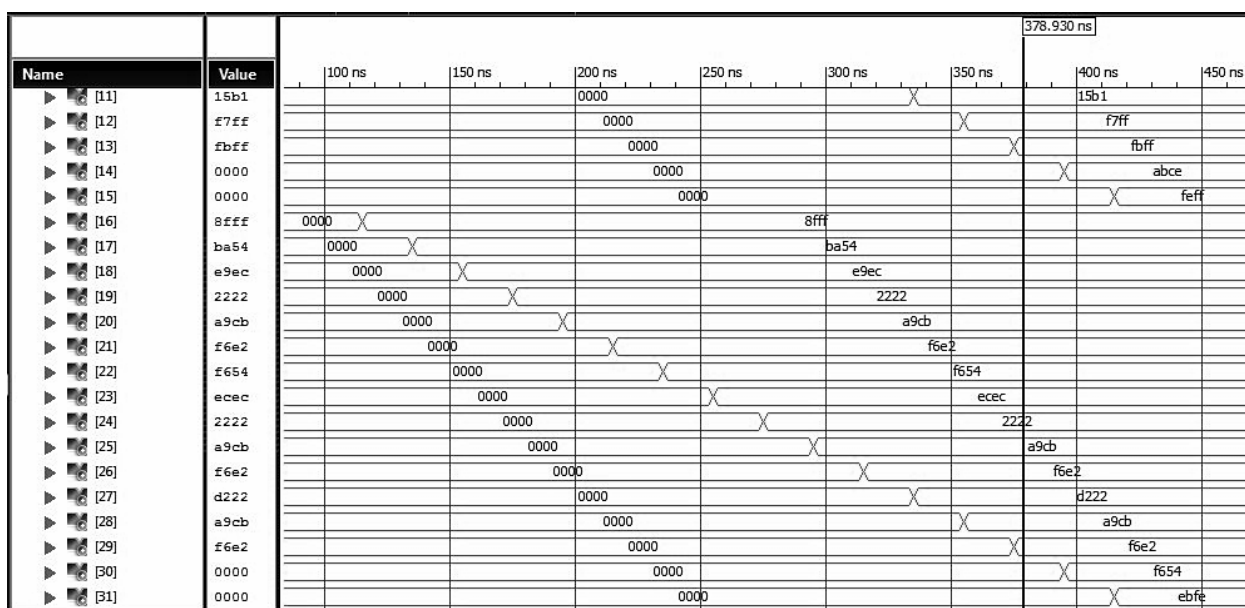


Рисунок 4.19 – Моделювання стану регістрів

Реалізація всіх векторних операцій блоку логічних обчислень, виконуваних з тактовою частотою 100 МГц, для одного УПУ в середовищі Verilog з наступною післясинтезною реалізацією в кристалі програмувальної



логіки Virtex 6, Xilinx містить 2400 еквівалентних вентилів. Мультипліціювання блоку логічного секвенсора до 16 штук зажадало  $38400+5200=43600$  вентилів, серед яких 5200 вентилів необхідні для реалізації обміну інформації між секвенсорами мультипроцесора.

Графіки порівняльних характеристик тестування (діагностування) цифрових блоків до й після впровадження результатів дисертаційної роботи представлені на рис. 4.20 – 4.21. Перший з них ілюструє істотне (28%) зменшення обсягу діагностичної інформації для 9 модулів за рахунок вибіркості тестів відносно асерційних моніторів для діагностування функціональних порушень. Другий показує зменшення загального часу тестування цифрових модулів, що у середньому становить 47%. При цьому час діагностування – пошук причини й місця дефекту – займає менш, ніж 0,1 частина від часу тестування цифрових систем на кристалах.

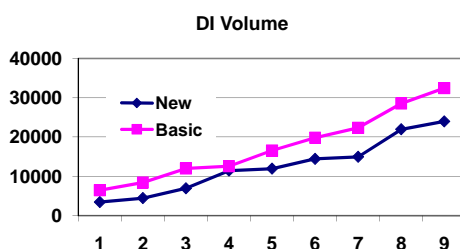


Рисунок 4.20 – Аналіз обсягів діагностичної інформації

На рис 4.20 і 4.21 представлені графіки порівняльного аналізу ефектів, отриманих у результаті впровадження моделей і методів діагностування компонентів цифрових систем на кристалах. При цьому діагностичні експерименти були виконані на 4 блоках Case Study і на п'яти моделях-

прототипах промислових зразків, що представляють собою IP-cores DSP-перетворювачів.

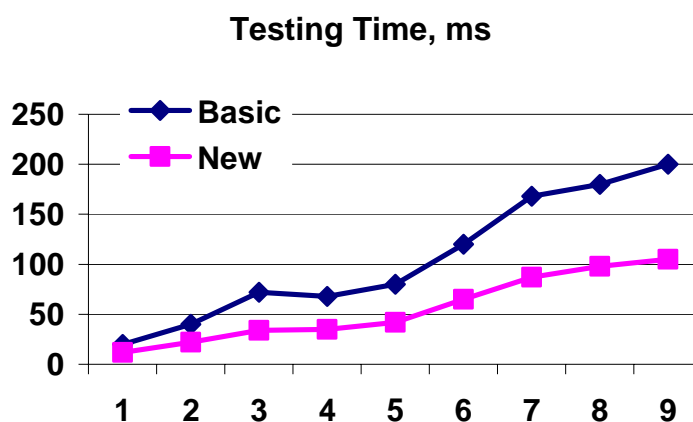


Рисунок 4.21 – Аналіз часу тестування So-компонентів

Порівнянню підлягали обсяги діагностичної інформації базових і нової матричних моделей для пошуку дефектів, а також загальний час тестування функціональностей на тих самих Testbench. В останньому випадку ефект (зменшення часу) отриманий за рахунок введення додаткової інфраструктури, рис. 4.22, до функціональності проекту, реалізованої на основі FPGA, що дозволяє вибірково здійснювати тестування й діагностування, а також перепрограмувати окремі модулі у випадку фіксації дефекту, завдяки введенню регістра граничного сканування стандарту IEEE 1500.

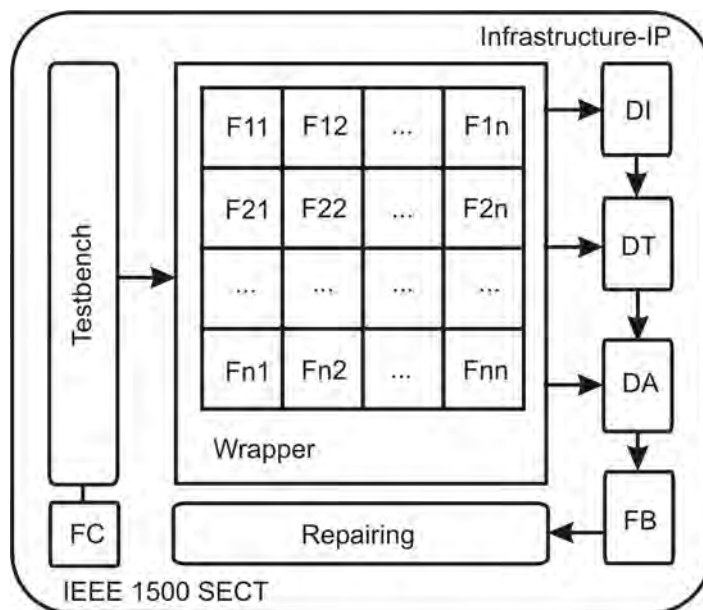


Рисунок 4.22 – Інфраструктура тестування SoC

Тут представлені блоки: Testbench – тести перевірки функціональних блоків, FC – функціональне покриття тесту, F – функціональні блоки пристрою, DI – діагностична інформація, DT – методи й засоби діагностування, DA – результати аналізу процесу діагностування, FB – несправні функціональні модулі пристрою, Repairing – відновлення працездатності функціональних модулів. Сервісне обслуговування окремого осередку функціональності здійснюється за допомогою осередку граничного сканування, представленої на рис 4.23.

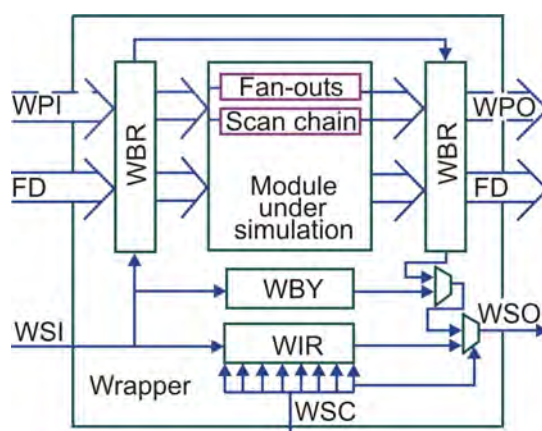


Рисунок 4.23 – Осередок граничного сканування

## 4.6 Висновки

Існуючі програмні аналоги практично не пропонують асоціативно-логічних технологій пошуку, розпізнавання й прийняття рішень у дискретному інформаційному просторі [89, 90, 96]. Практично всі вони використовують універсальну систему команд сучасного дорогого процесора з математичним співпроцесором. У той же час, апаратні спеціалізовані засоби логічного аналізу, що є їхніми прототипами [62, 63, 91], як правило, орієнтовані на побітову або неекторну обробку інформації. Запропонований новий підхід векторно-логічної обробки асоціативних даних з повним виключенням арифметичних операцій, що впливають на швидкодію й апаратну складність, може бути ефективно реалізований на основі використання сучасної мікроелектронної апаратури у вигляді мультипроцесорної цифрової системи на кристалі. Фактична реалізація підходу заснована на пропозиції моделей і методів, що використовують загальну ідею векторно-логічної метрики кіберпростору:

1. Процес-моделі аналізу асоціативних таблиць орієнтовані на досягнення високої швидкодії аналізу інформаційних об'єктів і підрахунку критеріїв якості їхньої взаємодії на основі векторних логічних операцій для пошуку, розпізнавання образів, прийняття й оцінювання рішень у кіберпросторі.

2. Метод паралельного розв'язок асоціативно-логічних завдань із мінімальним числом векторних логічних операцій і повним виключенням арифметичних команд, що забезпечує високу швидкодію, мінімальну вартість і незначне енергоспоживання обчислювача, реалізованого на кристалі програмувальної логіки.

3. Нові векторно-логічні процеси-моделі убудованого діагностування цифрових систем на кристалах, пошуку квазіоптимального покриття, що використовують засоби логічного асоціативного мультипроцесора, паралельні операції обчислювальних процесів і векторно-логічний критерій якості.

4. Розроблено архітектуру мультиматричного процесора, орієнтованого на підвищення швидкодії процедур убудованого діагностування несправностей, що відрізняється використанням паралельних логічних векторних операцій `and`, `or`, `xor`, `slc`, що дає можливість істотно ( $\times 10$ ) підвищити швидкодію діагностування одиночних і/або кратних дефектів по таблиці активізації функціональних модулів.

5. Запропоновано інфраструктуру діагностування апаратних блоків цифрових систем на кристалах, що включає чотири процеси-моделі для тестування, діагностування, мінімізації простору несправностей і виправлення помилок, замкнуті в цикл, що дає можливість зменшити час відновлення працездатності.

6. Розроблено апаратні модулі паралельного виконання інструкцій процесора, які дають можливість одночасної реалізації чотирьох логічних операцій над умістом таблиць, що на порядок підвищує швидкодію процесів діагностування функціональних блоків.

7. Синтезовано регістровий операційний модуль виконання векторних логічних операцій, використовуваних при діагностуванні й мінімізації потужності підозрюваних несправностей для убудованої інфраструктури тестування, що дає можливість підвищити швидкодію процесу пошуку дефектів у цифрових системах на кристалах.

8. Практична реалізація моделей і методів тестування використовує середовище моделювання Riviera компанії Aldec для створення діагностичної інформації. Уведені в систему модулі діагностування вдосконалили існуючий процес тестування, що дало можливість на 3% підвищити якість цифрового виробу.

9. Практична значимість отриманих результатів підтверджена створенням убудованого компонента для діагностування й відновлення працездатності пам'яті в цифровій системі на кристалі. Подальші дослідження спрямовані на розробку прототипу логічного асоціативного мультипроцесора для розв'язку задач пошуку, розпізнавання та прийняття рішень за допомогою векторного логічного аналізу.

## 5 МЕТОД ПІДВИЩЕННЯ ТЕСТОПРИДАТНОСТІ ПРОЦЕСОРНИХ СИСТЕМ З ВИКОРИСТАННЯМ ФУНКЦІОНАЛЬНИХ ЕЛЕМЕНТІВ НА БАЗІ АРИФМЕТИЧНИХ ОПЕРАЦІЙ

5.1 Математичний апарат методу підвищення тестопридатності процесорних систем

У даному розділі досліджується математичний апарат підвищення тестопригодності критичних систем керування, шляхом розробки додаткових моделей і методів тестового й функціонального діагностування на базі використання багаторозрядного подання вхідних даних і методів їхньої обробки в логічних схемах.

Об'єкт дослідження – критична система (КС) взаємодіючих програмно-апаратних компонентів. Всі стани КС, з урахуванням всіх типів її функціональних порушень (ФН) не виводять об'єкт за межі заданої функціональності, представлені булевою функцією в момент часу (такт) і:

$$Y_i = f(X_{1i}, X_{2i}, \dots, X_{ji}, \dots, X_{ni}, S_{i-1}), X_i, S_{i-1}, Y_i \in R = \{0, \dots, 1\}$$

Модель ФН накладається на графову структуру функціональних модулів (рис. 5.1), що мають вхідні й вихідні транзакційні змінні. Транзакційний граф представлений дугами – функціональностями (сервісами) з моніторами (асерціями), а також вершинами, що формують стан КС, за допомогою змінних, пам'яті, інтерфейсних портів вводу-виводу інформації, приємопередавачів, терміналів, комп'ютерів:  $F = (A * B) \times S$ , де  $S = \{S_1, S_2, \dots, S_i, \dots, S_m\}$  – вершини або стани КС при моделюванні тестових сегментів. Кожний стан  $S_i = \{S_{i1}, S_{i2}, \dots, S_{ij}, \dots, S_{ip}\}$  визначається значеннями істотних змінних КС (змінні, пам'ять, термінали, комп'ютери).

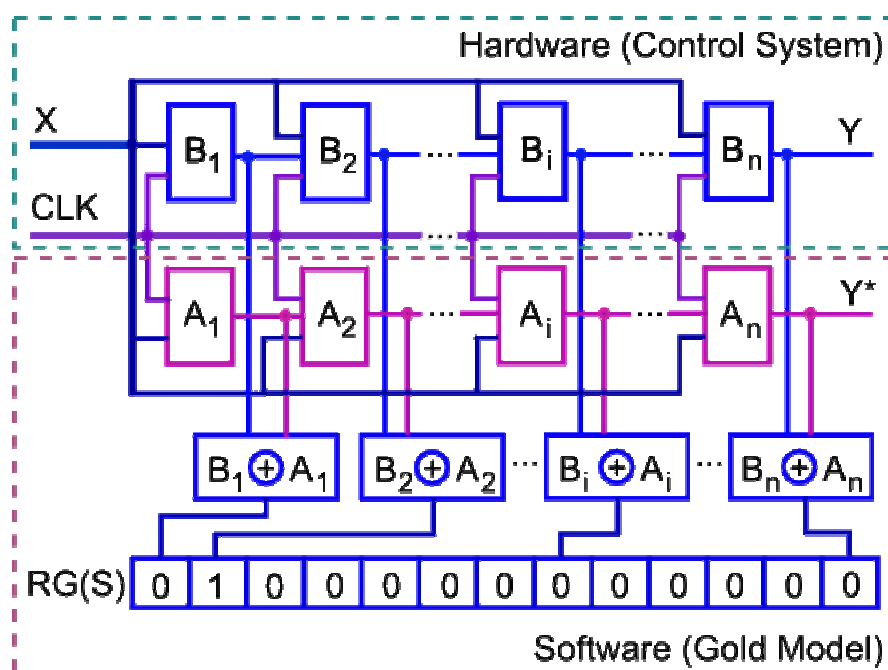


Рисунок 5.1 – Схема діагностування функціональних модулів КС

Кожному з функціональних блоків:

$$B = (B_1, B_2, \dots, B_i, \dots, B_n), \quad \bigcup_{i=1}^n B_i = B; \quad \bigcap_{i=1}^n B_i = \emptyset,$$

може бути поставлена у відповідність асерція  $A_i \in A = \{A_1, A_2, \dots, A_i, \dots, A_n\}$  для моніторингу функціональностей у часі й у просторі.

Метрика – спосіб виміру відстані в просторі між компонентами процесів або явищ, описаних векторами логічних змінних. Відстань (булева похідна, ступінь зміни, розходження або близькості) визначається хог-відношенням векторів (матриць), що позначають компоненти процесу або явища, що відрізняє його від кодової відстані за Хемінгом. Процедури порівняння, виміру, оцінювання, розпізнавання, тестування, діагностування, оперують хог-відношенням об'єктів або їхніх компонентів. Компонент простору

представлений  $k$ -мірним вектором  $a = (a_1, a_2, \dots, a_j, \dots, a_k)$ ,  $a_j \in R = \{0, \dots, 1\}$ , де кожна його координата визначена в многоразрядном алфавіті,  $[0;1)$  – «неправда»,  $[1]$  – «істина». Нуль-вектор є  $k$ -мірний кортеж, всі координати якого дорівнюють нулю:  $a_j = 0, j = \overline{1, k}$ . Метрика  $\beta$  кібернетичного простору визначається рівністю:

$$\beta = \bigoplus_{i=1}^n d_i = 0,$$

яке формує нуль-вектор для хог-суми відстаней  $d_i$  між ненульовим і кінцевим числом об'єктів, замкнених у цикл. Тут  $n$  – кількість відстаней між компонентами (векторами) простору, що становлять цикл  $D = (d_1, d_2, \dots, d_i, \dots, d_n)$ ,  $d_i$  – є вектор відстані, що відповідає ребру циклу, що з'єднує два компоненти (вектора)  $a, b$  простору, що далі позначається без індексу як  $d(a, b)$ . Відстань між двома об'єктами  $a$  і  $b$  є похідний вектор:  $d(a, b) = (a_j \oplus b_j)_1^k$ . Векторному значенню відстані відповідає норма (скаляр), обумовлена кодовою відстанню за Хемінгом між двома векторами у вигляді числа одиниць вектора  $d(a, b)$ .

Повний збіг двох об'єктів характеризується нульовою оцінкою їхньої розбіжності. Протилежним варіантом оцінювання є максимальна розбіжність відповідна 1. При цьому, всі інші проміжні варіанти розбіжності оцінюються в діапазоні  $(0;1)$ . Достоїнство введеного критерію розбіжності (неприналежності, розходження) полягає в лінійності зміни його чисельного значення від 0 до 1 у міру збільшення «відстані» від повного збігу двох об'єктів до максимально можливого, що дозволяє здійснювати тестування на значно більшій кількості комбінацій вхідних даних і підвищити тестопридатність КС.

Проблема синтезу або аналізу компонентів довільної КС може бути сформульована у вигляді взаємодії (симетричної різниці – аналог хог-операції



на булеані) у кібернетичному просторі її моделі  $F$  із входніми впливами  $T$  і реакціями  $L$ :

$$f(F, T, L) = \emptyset \rightarrow F \Delta T \Delta L = \emptyset .$$

У свою чергу, процедури синтезу тестів, моделювання ФН і їхнього діагностування можна звести до хог-відношень на графі (рис. 5.2) повної взаємодії чотирьох вершин (F-функціональність (модель), U – КС, T – входні впливи, L-L – реакція на ФН)  $G = \{F, U, T, L\}$ .

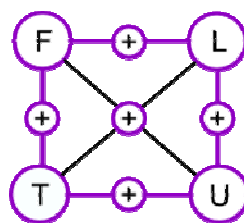


Рисунок 5.2 – Граф взаємодії компонентів тестування

## 5.2 Модифікація дедуктивного методу пошуку несправностей

Сутність дедуктивного методу полягає в порівнянні входних і вихідних даних КС із метою виявлення ФН. При цьому виконуються процедури порівняння інформації на виході системи у свідомо штатному (функціональному) режимі й у ситуації, що викликає підозру. Для імплементації методу в апаратуру захистів необхідно мати графову модель логіки її функціонування, що досить просто може бути перетворена в систему логічних рівнянь, придатну для дедуктивного аналізу. Далі пропонується модель дедуктивно-паралельного синхронного аналізу КС, що дозволяє за одну ітерацію обробки структури виявити всі ФН, що перевіряються на тест-векторі. Мета дедуктивного аналізу полягає у визначенні якості синтезованого тесту щодо повноти покриття їм ФН і побудові таблиці перевірки тестовими

наборами всіх виявлених ФН КС для виконання процедур діагностування. Така модель заснована на рішенні рівняння:

$$L = T \oplus F, \quad (5.1)$$

де  $F = (F_{m+1}, F_{m+2}, \dots, F_i, \dots, F_n) (i = m+1, n)$  – сукупність функцій справної (коректної) поведінки КС;  $m$  – число входів;  $Y_i = F_i(X_{i1}, \dots, X_{ij}, \dots, X_{in_i})$  –  $n_i$ -входою  $i$ -й функціональний елемент схеми, що реалізує функцію  $F_i$  для визначення стану лінії (виходу)  $Y_i$  на тест-векторі  $T_t$ ; тут  $j$ - $X_{ij}$ -й вхід  $i$ -го функціонального елемента; тест  $T = (T_1, T_2, \dots, T_t, \dots, T_k)$  – упорядкована сукупність багаторозрядних векторів, довизначення в процесі справного моделювання на множини вхідних, внутрішніх і вихідних ліній, об'єднана у двовимірну матрицю

$$T = [T_{ti}] = \begin{bmatrix} T_{11}, T_{12}, \dots, T_{1i}, \dots, T_{1n} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ T_{t1}, T_{t2}, \dots, T_{ti}, \dots, T_{tn} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ T_{k1}, T_{k2}, \dots, T_{ki}, \dots, T_{kn} \end{bmatrix}, \quad (5.2)$$

для якої не вхідна координата визначається шляхом моделювання функції  $T_{ti} = Y_i = F_i(X_{i1}, \dots, X_{ij}, \dots, X_{in_i})$  на тест-векторі  $T_t$ ;  $L = (L_1, L_2, \dots, L_t, \dots, L_k)$  – множина дедуктивних моделей, обумовлених виразом (5.1), де  $L_t = (L_{t1}, L_{t2}, \dots, L_{ti}, \dots, L_{tn})$ ;

$$L_{ti} = T_t \oplus F_i \quad (5.3)$$

– дедуктивна функція (ДФ) паралельного моделювання несправностей на тест-векторі  $T_t$ , що відповідає справному елементу  $F_i$ , що дає можливість обчислювати список вхідних ФН, що транспортуються на вихід елемента  $F_i$ .

Поняття синхронності моделі (5.1) визначається умовою:  $\Delta t = (t_{j+1} - t_j) \gg \tau \gg \tau_i$ , коли інтервал часу між зміною вхідних наборів  $(t_{j+1} - t_j)$ , що подаються на КС, набагато більше максимальної затримки  $\tau$  й елемента  $\tau_i$ . Це дозволяє виключити час як несуттєвий параметр, що використовується в технологіях моделювання й синтезу тестів. У загальному випадку, коли функція КС представлена таблицею істинності, застосування формули (5.1) дозволяє одержати для заданого тест-вектора  $T_t$  таблицю транспортування ФН, по якій можна записати ДФ моделювання несправностей. Приклади одержання таких функцій для елемента «2 з 3-х» представлені в наступному виді (перший доданок – тест-вектор, другий й результат – таблиці істинності й транспортування ФН):

$$\begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & L_1 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$L_1 = X_1 X_3;$$

$$\begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & 1 & 0 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & L_1 \\ \hline 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ \hline \end{array}$$

$$L_2 = X_1(X_2 \oplus X_3).$$

Дедуктивні функції  $L_1, L_2$  записані у вигляді диз'юнктивної нормальної форми по конститuentам одиниці таблиць транспортування ФН.

Приклади одержання дедуктивних функцій для елемента «2 з 4-х» наведені нижче.

$$\begin{array}{|c|c|c|c|c|} \hline X_1 & X_2 & X_3 & X_4 & Y \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|c|} \hline X_1 & X_2 & X_3 & X_4 & Y \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline X_1 & X_2 & X_3 & X_4 & L_1 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$L_1 = X_1 X_4 (\bar{X}_2 \vee X_2 X_3);$$

$$\begin{array}{|c|c|c|c|c|} \hline X_1 & X_2 & X_3 & X_4 & Y \\ \hline 1 & 1 & 0 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|c|} \hline X_1 & X_2 & X_3 & X_4 & Y \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline X_1 & X_2 & X_3 & X_4 & L_1 \\ \hline 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}$$

$$L_2 = X_1 X_2 X_4.$$

При розбивці тесту на складові вектори рівняння (5.1) одержання ДФ для  $T_t \in T$  приймає наступний вид:  $L_t = T_t \oplus F$ . Якщо функціональний опис КС представлений компонентами (примітивами), що формують стан всіх ліній (з'єднань), то як формула перетворення справної моделі примітива  $F_i$  на тест-векторі  $T_t$  в дедуктивну функцію  $L_{ti}$  виступає наступний вираз:

$$\begin{aligned} L_{ti} = T_t \oplus F_i = f_{ti}[(X_{i1} \oplus T_{t1}), (X_{i2} \oplus T_{t2}), \dots, \\ (X_{ij} \oplus T_{tj}), \dots, (X_{in_i} \oplus T_{tn_i})] \oplus T_{ti}, \end{aligned} \quad (5.4)$$

яке є основою дедуктивного аналізу ФН блоку захистів.

Модифікований дедуктивний метод пошуку несправностей дозволяє виявити не тільки константні несправності, але й функціональні порушення логічних елементів з використанням арифметичних операцій.

Приклад. Одержати дедуктивну функцію паралельного моделювання ФН на вичерпному тесті для функціонального елемента «2 з 3-х». З урахуванням виразу (5.4) виконуються наступні очевидні перетворення для функції «2 з 3-х»:

$$\begin{aligned}
 & L_{2\text{from}3}[T=(000,011,101,111,110,100,010,001), F=(X_1X_2 \vee X_1X_3 \vee X_2X_3)] = \\
 & = L\{\bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee \\
 & \vee \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3) \wedge [(X_1 \oplus T_{t1})(X_2 \oplus T_{t2}) \vee (X_1 \oplus T_{t1})(X_3 \oplus T_{t3}) \vee \\
 & \vee (X_2 \oplus T_{t2})(X_3 \oplus T_{t3})] \oplus T_{t4}\} = (\bar{x}_1\bar{x}_2\bar{x}_3)\{[(X_1 \oplus 0)(X_2 \oplus 0) \vee (X_1 \oplus 0)(X_3 \oplus 0) \vee \\
 & \vee (X_2 \oplus 0)(X_3 \oplus 0)] \oplus 0\} \vee (\bar{x}_1x_2x_3)\{[(X_1 \oplus 0)(X_2 \oplus 1) \vee (X_1 \oplus 0)(X_3 \oplus 1) \vee \\
 & \vee (X_2 \oplus 1)(X_3 \oplus 1)] \oplus 0\} \vee (x_1\bar{x}_2x_3)\{[(X_1 \oplus 1)(X_2 \oplus 0) \vee (X_1 \oplus 1)(X_3 \oplus 1) \vee \\
 & \vee (X_2 \oplus 0)(X_3 \oplus 1)] \oplus 1\} \vee (x_1x_2x_3)\{[(X_1 \oplus 1)(X_2 \oplus 1) \vee (X_1 \oplus 1)(X_3 \oplus 1) \vee \\
 & \vee (X_2 \oplus 1)(X_3 \oplus 1)] \oplus 1\} \vee (x_1x_2\bar{x}_3)\{[(X_1 \oplus 1)(X_2 \oplus 1) \vee (X_1 \oplus 1)(X_3 \oplus 0) \vee \\
 & \vee (X_2 \oplus 1)(X_3 \oplus 0)] \oplus 1\} \vee (x_1\bar{x}_2\bar{x}_3)\{[(X_1 \oplus 1)(X_2 \oplus 0) \vee (X_1 \oplus 1)(X_3 \oplus 0) \vee \\
 & \vee (X_2 \oplus 0)(X_3 \oplus 0)] \oplus 0\} \vee (\bar{x}_1x_2\bar{x}_3)\{[(X_1 \oplus 0)(X_2 \oplus 1) \vee (X_1 \oplus 0)(X_3 \oplus 0) \vee \\
 & \vee (X_2 \oplus 1)(X_3 \oplus 0)] \oplus 0\} \vee (\bar{x}_1\bar{x}_2x_3)\{[(X_1 \oplus 0)(X_2 \oplus 0) \vee (X_1 \oplus 0)(X_3 \oplus 1) \vee \\
 & \vee (X_2 \oplus 0)(X_3 \oplus 1)] \oplus 0\} = \bar{x}_1 \vee \bar{x}_2\bar{x}_3X_2X_3 \vee x_1X_3(X_1 \vee \bar{x}_2x_3\bar{X}_1) \vee \\
 & \vee x_1\bar{x}_3X_2(\bar{x}_2\bar{X}_1 \vee \bar{X}_3).
 \end{aligned}$$

Тут  $T_t = (T_{t1}, T_{t2}, T_{t3}, T_{t4}), (t = \bar{1}, \bar{8})$  – тест-вектор, що має 4 координати, де остання з них визначає стан виходу трьохходового елемента «2 з 3-х».

В аналітичній моделі представлені булеві  $(x_1, x_2)$  і регістрові  $(X_1, X_2)$  для кодування ФН входи, змінна вибору типу справної функції (AND, OR), вихідна регістрова змінна  $Y$ . Стани входів  $x_1, x_2$  і змінна вибору елемента визначають одну із чотирьох дедуктивних функцій для одержання вектора  $Y$  несправностей, що перевіряються. Для апаратної реалізації дедуктивно-паралельного моделювання на основі симулятора фірми Aldec, USA може бути використана обчислювальна структура, представлена на рис. 5.3.

Особливість схемної реалізації полягає у спільному виконанні емуляції функцій логічних елементів And, Or і обробки багаторозрядних векторів

несправностей шляхом виконання операцій логічного множення, заперечення й додавання.

Функціональне призначення основних блоків (пам'ять і процесор):

1.  $M = [M_{ij}]$  – квадратична матриця моделювання ФН, де  $i, j = 1, q$ ;  $q$  – загальне число ліній в оброблюваній КС.
2. Вектори збереження станів коректного моделювання, певні в моменти часу  $t-1$  і  $t$ , необхідні для формування дедуктивних функцій примітивів.
3. Модуль пам'яті для зберігання опису КС у вигляді структури логічних елементів.
4. Буферні регістри, розмірністю  $q$ , для зберігання операндів і виконання регістрових паралельних операцій над векторами ДП, лічених з матриці  $M$ .
5. Дедуктивно-паралельний симулятор, що обробляє за один такт дві регістрових змінних  $X1$ ,  $X2$  з метою визначення вектора ДП, що транспортуються на вихід логічного елемента  $Y$ .

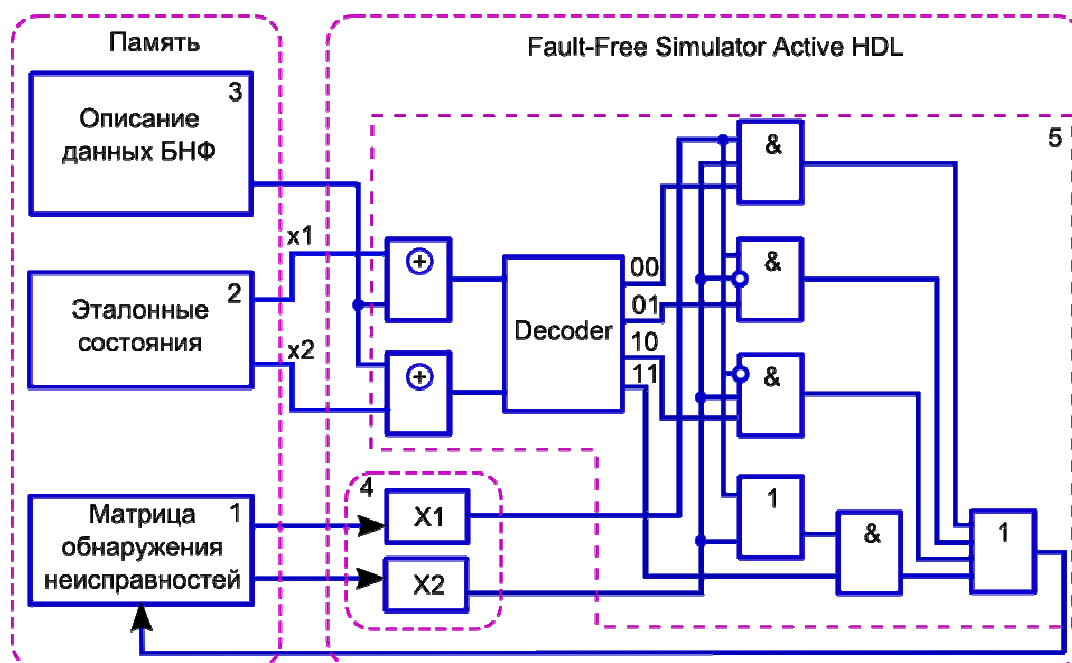


Рисунок 5.3 – Структура апаратного моделювання

### 5.3 Аналіз тестопридатності

Розроблений математичний апарат показує, що загальним підходом, який забезпечує підвищення тестопридатності устаткування, є використання базових логічних елементів («і», «або», «2 з 3» і ін.) із багаторозрядним поданням вхідних сигналів і логіки їхньої обробки, оскільки кількість різних комбінацій вхідних сигналів збільшується при цьому в статичній залежності в порівнянні з використанням традиційної бінарної логіки. Одним з методів реалізації такого типу елементів є використання інтегральної оцінки (середнього арифметичного) значення від вхідних змінних. Наприклад, для елемента  $n$  – вхідного елемента «і» (AND) така реалізація має вигляд:

$$y = \frac{x_1 + \dots + x_n}{n},$$

$$x_1, \dots, x_n, y \in [0, \dots, 1].$$

Для дослідження тестопридатності при одиничних несправностях розглянемо аналітичні вирази моделюючої поведінки входів і виходів елемента AND в устаткуванні захистів (A) і в устаткуванні діагностування (B):

$$y_A = \frac{x_{A1} + \dots + x_{Ai} + \dots + x_{An}}{n}; y_B = \frac{x_{B1} + \dots + x_{Bi} + \dots + x_{Bn}}{n}$$

При відсутності несправностей за входами і виходами маємо наступні залежності:

$$x_{A1} = x_{B1}; \dots; x_{Ai} = x_{Bi}; \dots; x_{An} = x_{Bn}; y_A = y_B$$

$$\Delta x_{A1} = \Delta x_{B1}; \dots \Delta x_{Ai} = \Delta x_{Bi}; \dots \Delta x_{An} = \Delta x_{Bn};$$

$$\Delta y_A = \Delta y_B = \frac{\Delta x_{A1} + \dots + \Delta x_{Ai} + \dots + \Delta x_{An}}{n} = \frac{\Delta x_{B1} + \dots + \Delta x_{Bi} + \dots + \Delta x_{Bn}}{n} \quad (5.5)$$

$$r_y = \Delta y_A - \Delta y_B = 0$$

Таким чином, при відсутності несправностей, неузгодженість  $r_y$  між виходами у в А та В буде дорівнювати 0.

З появою одиночної константної несправності за будь-яким входом і маємо наступні залежності:

$$\Delta x_{A1} = \Delta x_{B1}; \dots \Delta x_{Ai} \neq \Delta x_{Bi}; \dots \Delta x_{An} = \Delta x_{Bn};$$

$$\Delta y_A = \frac{\Delta x_{A1} + \dots + \Delta x_{Ai} + \dots + \Delta x_{An}}{n}; \Delta y_B = \frac{\Delta x_{B1} + \dots + \Delta x_{Bi} + \dots + \Delta x_{Bn}}{n};$$

$$r_y = \Delta y_A - \Delta y_B = \frac{\Delta x_{Ai} - \Delta x_{Bi}}{n} \neq 0.$$

З появою одиночної константної несправності по виходу маємо наступні залежності:

$$\Delta x_{A1} = \Delta x_{B1}; \dots \Delta x_{Ai} = \Delta x_{Bi}; \dots \Delta x_{An} = \Delta x_{Bn};$$



$$\Delta y_A \neq \Delta y_B = \frac{\Delta x_{A1} + \dots + \Delta x_{Ai} \dots + \Delta x_{An}}{n} = \frac{\Delta x_{B1} + \dots + \Delta x_{Bi} \dots + \Delta x_{Bn}}{n}; \quad (5.6)$$

$$r = \Delta y_A - \Delta y_B \neq 0.$$

З наведених вище аналітичних моделей (залежностей) треба, щоб застосування модифікації дедуктивного методу для багаторозрядної логіки дозволяє виявляти:

- одиничну несправність за входом елемента, з появою зміни сигналу на даному вході;
- одиничну несправність за виходом елемента, з появою зміни сигналу по кожному з  $n$  входів або їхньої комбінації.

Як слідує з (5.5) і (5.6) для комбінацій кратних несправностей за входами (або входам і виходом) можлива ситуація, коли несправності взаємно компенсують один одного щодо вихідного значення  $y$  в устаткуванні  $A$  і  $B$  і неузгодженість  $r$  при цьому дорівнює  $0$ , що є цілком прогнозованою властивістю для інтегральної схеми обробки.

Схема тестування логічного елемента на появу одиночної константної несправності за входом (тип несправності – відсутність зміни значення за входом  $x_1$  в  $A$  при його фактичній зміні на величину молодшого розряду АЦП), на прикладі AND3, наведена в табл. 5.1 і показує виявлення несправності за входом  $x_1$  при зміні значення вхідного сигналу (між тактами  $i$  та  $i+1$ ) на величину молодшого розряду АЦП (для спрощення взятий приклад 8-мі розрядного АЦП). У лівому стовпці показані розряди АЦП за трьома входами в устаткуванні  $A$  і  $B$ , у правому – вони ж, переведені в десятковий формат float (при цьому в дужках зазначені ці ж значення, наведені до діапазону 0-1), ненульове значення неузгодженості  $r$  у такті  $i+1$  між  $y$  в устаткуванні  $A$  і  $B$  вказує на виявлення несправності.

Результати дослідження, наведені в табл. 5.1 показують, що інтегральна обробка (середнє арифметичні) вхідних змінних дає тестопридатний результат для всіх випадків одиничних несправностей у логічному елементі, побудованому на багаторозрядній логіці. У випадку одиночної константної несправності за одним входом (змінною), для виявлення даної несправності досить однієї зміни сигналу за даним входом на будь-яку величину перевищуючи значення «молодшого» біта АЦП, при цьому несправність виявляється за неузгодженістю значення виходу логічного елемента у між устаткуванням захистів (А) і устаткуванням діагностування (В). Для виявлення несправності виходу процесора, пов'язаної з обробкою значень вхідних змінних досить зміни сигналу за кожним з n входів, у тому числі при штатній роботі устаткування за рахунок природної «флуктуації» багаторозрядних величин вхідних сигналів від датчиків.

Для випадків кратних несправностей, теоретично можлива ситуація, коли дві або більше несправності будуть компенсувати взаємовплив на підсумковий інтегральний результат, тобто неузгодженість значення виходу елемента у між устаткуванням захистів (А) і устаткуванням діагностування (В) буде дорівнювати 0, що є ознакою відсутності несправностей. Імовірність такого випадку може бути визначена виразом:

$$L = \frac{1}{2^k} * \lambda^2$$

де  $\lambda$  – інтенсивність потоку несправностей за одним входом в блоці елементів, що реалізує зазначену функцію, 1/год (для типових мікросхем оцінюється в діапазоні від  $10(-6)$  до  $10(-9)$ ); k-k – розрядність АЦП, шт.;  $\frac{1}{2^k}$  – дільник, що враховує багаторозрядне подання вхідних параметрів.

Таблиця 5.1 – Реакція виходів елемента AND3 в устаткуванні захистів А та в діагностичному встаткуванні В на зміну вхідного сигналу при константній несправності по даному входу

		Двійкове подання вхідних сигналів після АЦП (розрядність=k, приклад k=8). Значення $2^k - 1 = 2^8 - 1 = 255$ відповідає спрацьовуванню уставки		Десяткове подання у форматі float (десяткове подання, наведене до діапазону 0-1, тобто ділене на величину спрацьовування уставки $2^k - 1 = 2^8 - 1 = 255$ )	
		Такт i	Такт i+1	Такт i	Такт i+1
1		1001 0011	1001 0011	147,000 (0,5765)	147,000 (0,5765)
		1001 0011	1001 0010	147,000 (0,5765)	146,000 (0,5804)
2		1110 1110	1110 1110	238,000 (0,9333)	238,000 (0,9333)
		1110 1110	1110 1110	238,000 (0,9333)	238,000 (0,9333)
3		0111 1001	0111 1001	121,000 (0,4745)	121,000 (0,4745)
		0111 1001	0111 1001	121,000 (0,4745)	121,000 (0,4745)
		$y = \frac{x_1 + x_2 + x_3}{3}$		168,6667 (0,6614)	168,6667 (0,6614)
				168,6667 (0,6614)	168,3333 (0,6601)
				r=0 (0)	r=0,3334 (0,0013)
				Відсутність несправності	Виявлення несправності

Відповідно для типового АЦП із дозволом (значущі розряди) k=15, імовірність наявності взаємокомпенсованих несправностей на двох входах можна оцінити як:

$$L = \frac{1}{2^{15}} * \lambda^2$$

Даний вираз показує, що для мінімізації ймовірності таких збігів, необхідно мінімізувати кількість вхідних змінних у використовуваних

елементах, тобто не бажано використовувати складні елементи з більшою кількістю входів (і отже більшим значенням потоку несправностей  $\lambda$ ), а доцільніше розбити їх на більш дрібні елементи (мікросхеми). Кардинальним методом розв'язання даної проблеми може бути використання контролю не тільки виходу елементів, але й всіх внутрішніх змінних, тобто змінних, значення яких використовуються при розрахунку вихідного значення логічного елемента, таким чином, контролюються відповідні комірки пам'яті з яких вичитуються значення змінних при розрахунку виходу.

#### 5.4 Висновки

У даному розділі виконані наступні дослідження:

1. Розробка методу підвищення тестопридатності устаткування захистів УСБ за рахунок використання арифметичних операцій у функціональних елементах. Класифікація стандартних операцій (арифметичні, логічних, відношень), використовуваних при реалізації функціональних елементів різного призначення із складу устаткування захистів УСБ.

2. З урахуванням використання арифметичних операцій в устаткуванні захистів УСБ, виконаний аналіз рівнів «технологічного» і «електричного» шумів по вимірюваних параметрах, що підтверджує можливість контролю проходження будь-якої зміни значення вхідного сигналу в межах розв'язної здатності використовуваних АЦП від входу через всі функціональні елементи («порівняння з уставкой», «і», «або», «2з 4-х» і ін.), у яких бере участь даний сигнал, до дискретного елемента керування виконавчим механізмом.

3. За результатами аналізу зроблений висновок, про перевагу використання формату із плаваючої коми для реалізації подання значень вхідного, проміжного й вихідних змінних алгоритму в устаткуванні захистів УСБ. Аналіз виконаний шляхом оцінки мінімальної й достатньої точності для подання (виконання операцій) і окремо для відображення персоналу наступних

двох типів даних (при використанні АЦП із кількістю значущих розрядів 12 і 15 біт):

а) змінних, що характеризують стан вхідних параметрів у всьому діапазоні їхнього виміру (дані змінні використовуються в ланцюжку: нормування коду АЦП до діапазону виміру вхідного параметра – елемент «обмеження»- елемент «порівняння з уставкой»);

б) змінних, що характеризують стан логічних сигналів алгоритму в діапазоні від 0 до 1 (дані змінні використовуються для зв'язків між функціональними елементами, починаючи з виходу елемента «порівняння з уставкой»).

4. Удосконалено дедуктивний метод пошуку ФН у КС на базі зіставлення даних між КС і моделлю з метою виявлення ФН у штатні (функціональних) режимах роботи КС. Представлено структурну модель апаратного симулятора й пристрою моделювання в цілому, які орієнтовані на істотне підвищення швидкодії засобів моделювання КС великої розмірності шляхом поділу функцій коректного аналізу й обчислення списків що перевіряються ФН на тестових наборах.

## 6 ІНШІ РЕЗУЛЬТАТИ, ОТРИМАНІ ТА РЕАЛІЗОВАНІ В РАМКАХ НДР

### 6.1 Результативність виконання НДР

Таблиця 6.1 – Кількісні показники виконання НДР

№ з/п	Критерії	<u>Заплановано</u> (відповідно до запиту)	<u>Виконано</u> (за результатами НДР)	<u>%</u> <u>виконання</u>
		кількість	кількість	%
1.	<b>Публікації колективу виконавців НДР:</b>			
	1.1. Статті у журналах, що входять до наукометричних баз даних (*).	5	6	125%
	1.2. Публікації в матеріалах конференцій, що входять до наукометричних баз даних (*).	15	37	246%
	1.3. Статті у журналах, що включені до переліку наукових фахових видань України.	30	30	100%
	1.4. Публікації у матеріалах конференцій, тезах доповідей та виданнях, що не включені до переліку наукових фахових видань України.	30	35	116%
	1.5. Монографії, опубліковані за рішенням Вченої ради ВНЗ (наукової установи).	–	1 підготовлено	100%
	1.6. Підручники, навчальні посібники з грифом МОНмолодьспорт України (МОН України).	–	1 підготовлено	100%
	1.7. Навчальні посібники без грифу МОНмолодьспорт України (МОН України).	–	1 підготовлено	100%
1.8. Словники, довідники.	–	–	–	
2.	<b>Підготовка наукових кадрів:</b>			
	2.1. Захищено докторських дисертацій за тематикою НДР.	1	1 під керівництвом	100%
	2.2. Подано до розгляду у спеціалізовану вчену раду докторських дисертацій за тематикою НДР.	–	–	–
	2.3. Захищено кандидатських дисертацій за тематикою НДР.	3	5	170%
	2.4. Подано до розгляду у спеціалізовану вчену раду кандидатських дисертацій за тематикою НДР.	–	–	–
2.5. Захищено магістерських робіт за тематикою НДР.	9	28	311%	
3.	<b>Охоронні документи на об'єкти права інтелектуальної власності, які створено за тематикою НДР:</b>			
	3.1. Отримано патентів (свідоцтв авторського права) України.	1	1	100%

	3.2. Подано заявок на отримання патенту України.	–	–	–
	3.3. Отримано патентів (свідоцтв авторського права) інших держав.	–	–	–
	3.4. Подано заявок на отримання патенту інших держав.	–	–	–
4.	<b>Участь з оплатою у виконанні НДР:</b>			
	4.1. Студентів.	–	–	–
	4.2. Молодих учених та аспірантів.	1	1	100%

## 6.2 Монографії та навчальні посібники

### Монографії:

1. Колективна монографія. Бондаренко М.Ф., Хаханов В.И., Энглезі И.П., Лобур М.В., Чумаченко С.В., Литвинова Е.И., Гузь О.А. Перспективные технологии XXI века «Облако мониторинга и управления дорожным движением – зеленая волна». Украина, Одесса. С. 80-100.

2. Хаханова Г.В., Бараннік В.В. Структурно-каскадні методи стиску та відновлення даних в телекомунікаційних та цифрових системах. Харьков: ХНУРЕ. 2013. 272 с. (підготовлено до друку).

### Підручники та посібники:

Хаханов В.І., Литвинова Є.І., Хаханова Г.В., Гузь О.О. «Проектування та верифікація цифрових систем на кристалах. Verilog та System Verilog». 2012. 528 с. (підготовлено до отримання грифу).

## 6.3 Патенти

Отримано патент на корисну модель – № 71361 Регістр зсуву. Зареєстровано 10.07.2012.

## 6.4 Наукові статті

- 1) Хаханов В.И., Гузь О.А., Ngene Christopher Umerah, Ольховой В. Процесс-модели анализа ассоциативных структур данных // АСУ и приборы автоматики. 2010. Вып. 152. С.21-26 (надруковано у 2011).
- 2) Хаханов В.И., Чумаченко С.В., Tiesoura Yves, Василенко В.А. Инфраструктура вычислительного процесса на основе векторных логических операций // АСУ и приборы автоматики. 2010. Вып. 153. С.4-12 (надруковано у 2011).
- 3) Хаханов В.И., Чумаченко С.В., Мищенко А.С., Зацарный А.В., Хаханова Ю.В. Инфраструктура диагностирования вредоносных программ в индивидуальном кибернетическом пространстве // АСУ и приборы автоматики. 2010. Вып. 153. С.19-32 (надруковано у 2011).
- 4) Хаханов В.И., Мищенко А.С., Вареца В.В. Метрика алгебры векторной логики для кибернетического пространства // Радиоэлектроника и информатика. 2010. №3. С. 39-42 (надруковано у 2011).
- 5) Хаханов В.И., Тиекура Ив (Tiesoura Yves). Логический метод поиска дефектов по таблице неисправностей // Радиоэлектроника и информатика. 2010. №3. С. 43-45 (надруковано у 2011).
- 6) \*Хаханов В.И., Литвинова Е.И., Чумаченко С.В., Гузь О.А. Логический ассоциативный вычислитель // Электронное моделирование. 2011. № 1(33). С. 73-89. [Индексується: Cambridge Scientific Abstracts, Computer and Information Systems Abstracts, INIS Collection, Inspec, ВИНТИ РАН].
- 7) Хаханов В.И., Чумаченко С.В., Литвинова Е.И., Гузь О.А. Инфраструктура диагностирования программно-аппаратных систем // Радиоэлектроника. Информатика. Управление. Запорожье ЗНТУ.2011. №1(24) С.129-138.
- 8) Хаханов В.И., Зайченко С.А., Чумаченко С.В., Литвинова Е.И., Гузь О.А. Диагностирование модулей программно-аппаратных систем //



- Інформаційно-керуючі системи на залізничному транспорті. 2011. № 4. С. 47-55.
- 9) Тиекура Ив (Tiesoura Yves), Чумаченко С.В., Хаханов В.И. Актуальные проблемы анализа киберпространства // АСУ и приборы автоматики. 2011. Вып. 154. С.59-76.
  - 10) Ив Тиекура (Yves Tiesoura), Хаханов В.И., Чумаченко С.В., Литвинова Е.И. Анализ киберпространства и диагностирование функциональных модулей // АСУ и приборы автоматики. 2011. Вып. 155. С.4-14.
  - 11) Хаханова А.В., Баранник В.В., Краснорущий А.А. Позиционное структурно-весовое кодирование бинарного представления трансформант // АСУ и приборы автоматики. 2011. Вып. 157. С.23-28.
  - 12) Hahanov V. East-West Design and Test Symposium // Design and test of Computers. 2011. USA. No 6. С. 134.
  - 13) Герасименко К.Е., Хаханов В.И. Повышение тестопригодности критических систем управления // АСУ и приборы автоматики. 2011. Вып. 157. С.107-116.
  - 14) Хаханова А.В., Баранник В.В., Кривонос В.Н. Кодирование значимых компонент трансформант // Радиоэлектроника и информатика. 2012. №2. С. 32-35.
  - 15) Murad Ali Abbas, Хаханов В.И., Литвинова Е.И., Хаханова И.В. Инфраструктура встроенного восстановления логических PLD-схем // Радиоэлектроника и информатика. 2012. №2. С. 54-57.
  - 16) Ngene C.U. The Essentials of Testing Digital Circuits // Radioelektroniks and informatics. 2012. № 1. P. 22-30.
  - 17) Горобец А.А., Мурад Али Аббас, Хаханов В.И., Литвинова Е.И., Чумаченко С.В. Квантовые технологии тестирования и ремонта цифровых систем на кристаллах // АСУ и приборы автоматики. 2012. Вып. 158. С.18-35.
  - 18) Хаханова И.В. TLM-модель процессора анализа логических отношений // АСУ и приборы автоматики. 2012. Вып. 158. С.48-55.

- 19) Мурад Али Аббас, Дахири Фарид, Багхдади Аммар Авни Аббас, Хаханов В.И., Литвинова Е.И. Технологии восстановления работоспособности мультипроцессорных систем на кристаллах // АСУ и приборы автоматики. 2012. Вып. 159. С.4-14.
- 20) Murad Ali Abbas, Хаханов В.И., Литвинова Е.И., Хаханова И.В. Модели встроенного ремонта логических блоков // Радіоелектроніка, інформатика, управління. 2012. №2. С. 122-129.
- 21) Хаханов В.И., Герасименко К.Е. Обоснование контролепригодности многоразрядной логики защит в критических системах управления // АСУ и приборы автоматики. 2012. Вып. 160. С.54-59.
- 22) Хаханова И.В. Синтез кубитных моделей логических функций // АСУ и приборы автоматики. 2012. Вып. 161. С.4-9.
- 23) Сыревич Е.Е., Шкиль А.С., Альмадхоун С., Фастовец Г.П. Структурное и функциональное диагностирование HDL-моделей цифровых устройств в САПР РЭА // Інформаційно-керуючі системи на залізничному транспорті. 2013. № 2. С. 75-82.
- 24) Акимов Р.И., Хаханова А.В. Модель оценки информативности биадического представления межкадровых апертур для последовательности Р-кадров // АСУ и приборы автоматики. 2013. Вып. 162. С.35-40.
- 25) Баранник В.В., Кривонос В.Н., Хаханова А.В. Метод компактного представления вектора значимых компонент трансформант // АСУ и приборы автоматики. 2013. Вып. 162. С.40-47.
- 26) Стенли Гайдук, Владимир Хаханов. Структура логического ассоциативного мультипроцессора // АСУ и приборы автоматики. 2013. Вып. 162. С.4-21.
- 27) Изотов А.С., Немченко В.П. Анализ конформности сетевых протоколов // Інформаційно-керуючі системи на залізничному транспорті. 2013. №4 (приложение). С. 55-57.

- 28) Филиппенко И.В. Проектирование систем радиочастотной идентификации с использованием широкополосной технологии // Вестник НТУ «ХПИ». 2013. № 70(10-43). С. 59-65.
- 29) Мова А.Ю., Усиченко Р.И., Кривуля Г.Ф., Бабич А.В. Пакет DNR переменной длины для модели обратной связи RTCP с диагностическим узлом // Інформаційно-керуючі системи на залізничному транспорті. 2013. №4. С. 56-61.
- 30) Сыревич Е., Власов И., Павлов О., Кривуля Г. Особенности применения наномемристорной логики для проектирования цифровых систем // Вестник Херсонского национального университета. 2013. №1(46). С.284-290.
- 31) Hahanova I.V. Coverage Problem Solving on Quantum Computing // Radioelektroniks and informatics. 2012. № 4. P.62-67.
- 32) \*Хаханов В.И., Литвинова Е.И. Мультипроцессор для анализа информационного пространства. 1. Архитектура логического ассоциативного мультипроцессора // Вестник Томского государственного университета. №1. 2011. С. 95-108 [входит в систему Российского Индекса Научного Цитирования (РИНЦ) на платформе <http://elibrary.ru>].
- 33) \*Хаханов В.И., Чумаченко С.В. Мультипроцессор для анализа информационного пространства. 2. Инфраструктура векторно-логического анализа // Вестник Томского государственного университета. №2. 2011. С. 108-116 [входит в систему Российского Индекса Научного Цитирования (РИНЦ) на платформе <http://elibrary.ru>].
- 34) \*Hahanov V., Wajeb Gharibi, Litvinova E., Chumachenko S. Information analysis infrastructure for diagnosis // Information an international interdisciplinary Journal. 2011. Vol. 14. № 7. P. 2419-2433 [Indexed by Scopus, JDream, Mathematical Reviews, Zentralblatt MATH, ProQuest, Swets, EBSCO].
- 35) \*Кучеренко Д.Е., Шкиль А.С., Кривуля Г.Ф. Анализ продукционных правил в экспертных системах диагностирования // Рига. Автоматика и вычислительная техника. 2013. №6. С.59-72.

- 36) \*Kucherenko D. Ye., Krivoulya G.F., Shkil' A. S. Production Rules in Expert Systems of Diagnosis // Analysis of automatic control and computer sciences. 2013. Vol.47. No. 6. P.331-341.

#### 6.5 Тези доповідей на конференціях

- 1) Мищенко А.С. (научный руководитель д.т.н., проф. Чумаченко С.В.). Стандартизация Cyber Space // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.122-123.
- 2) Войный И.С, Дахири Ф.Н., Зенович И.А. (научный руководитель – ст. пр. Зайченко С.А.). Комплексная верификация пересечений тактовых доменов в моделях цифровых систем // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.60-61.
- 3) Горшков А.Н. (научный руководитель – асс. Обризан В.И.). Исследование способов передачи символьной и графической информации на удаленный терминал // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.64-65.
- 4) Минская А.Г., Потерба В.С., Арефьев А.А. (научный руководитель – ст. пр. Зайченко С.А.) Метод ограниченного псевдослучайного тестирования цифровых систем на основе взвешенных диаграмм решений // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.76-77.
- 5) Адамов А.С. (научный руководитель – д.т.н., проф. Хаханов В.И.). Угрозы безопасности персональных данных в социальных сетях // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.110-111.

- 6) Мищенко А.С., Зацарный А.В. (научный руководитель – д.т.н., проф. Хаханов В.И.). Инфраструктура диагностирования вредоносных программ в кибернетическом пространстве // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.118-119.
- 7) Мищенко А.С., Хаханова Ю.В. (научный руководитель – д.т.н. проф. Чумаченко С.В.) Использование облачных вычислений для решения проблем распределения нагрузки в социальных сетях // Материалы XV Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2011. часть 2. 18-20 апреля. С.120-121.
- 8) Хаханов В.И., Литвинова Е.И. Инфраструктура анализа и информационной безопасности киберпространства // Збірка матеріалів II Всеукраїнської науково-технічної конференції студентів, аспірантів та молодих вчених. 11-13 квітня 2011. Донецьк. С. 27-32.
- 9) Хаханов В.И., Чумаченко С.В., Мищенко А.С., Тиекура Ив. Эволюционная модель киберпространства // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии». 24-25 мая 2011.. С. 383-384.
- 10) Хаханов В.И., Чумаченко С.В., Адамов А.С. Инфраструктура программного продукта в Cyber Space // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии». 24-25 мая 2011. С. 385-386.
- 11) Хаханов В.И., Хаханова Ю.В., Щерба О.В. Мультиматричный процессор бинарных операций // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии». 24-25 мая 2011.. С. 387-388.

- 12) Хаханов В.И., Литвинова Е.И., Скворцова О.Б. Метрика кибернетического пространства // Сборник научных трудов VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии». 24-25 мая 2011. С. 389-390.
- 13) Litvinova E., Hahanova Y., Priymak A. Infrastructure for logic decision-making // Матеріали XI Міжнародної науково-технічної конференції CADSM 2011. 23-25 лютого. 2011. Львів. С. 75-78.
- 14) \*Ktiaman H., Hahanova A., Shcherbin D. High-speed technology for the solving logical problems // Матеріали XI Міжнародної науково-технічної конференції CADSM 2011. 23-25 лютого. 2011. Львів. С. 79-84 [Scopus].
- 15) \*Hahanov V., Chumachenko S., Mostovaya K. Metrics of vector logic algebra for cyber space // Матеріали XI Міжнародної науково-технічної конференції CADSM 2011. 23-25 лютого. 2011. Львів. С. 183-185 [Scopus].
- 16) \*Hahanov V., Tiesoura Yves, Pashchenko A., Olhovoy V. Algebra-logical method of fault diagnosis // Матеріали XI Міжнародної науково-технічної конференції CADSM 2011. 23-25 лютого. 2011. Львів. С. 186-187 [Scopus].
- 17) \*Gharibi W., Hahanov V., Litvinova E. Associative logical information analysis for cyber space // Матеріали XI Міжнародної науково-технічної конференції CADSM 2011. 23-25 лютого. 2011. Львів. С. 190-199 [Scopus].
- 18) \*Hahanov V., Guz O., Ngene Christopher Umerah. Process models for analyzing associative data structures // Матеріали XI Міжнародної науково-технічної конференції CADSM 2011. 23-25 лютого. 2011. Львів. С. 344-347 [Scopus].
- 19) \*Hahanov V., Litvinova E., Wajeb Gharibi. Infrastructure for analyzing information in a vector discrete space // First international workshop Criss-Dessert 2011. Kirovograd. May 11-13. С. 113-122 [Scopus].
- 20) \*Hahanov V., Wajeb Gharibi, Dong Won Park, Eugenia Litvinova. Cybercomputer for information space analysis // Proc. of IEEE East-West Design

- and Test Symposium. Sevastopol. 19-20 September 2011. P.66-71 [IEEE-Library, Scopus].
- 21) \*Hahanov V., Dong Won Park, Guz O., Priymak A., Galagan S. Verification and diagnosis of SoC HDL-code // Proc. of IEEE East-West Design and Test Symposium. Sevastopol. 19-20 September 2011. P.72-83 [IEEE-Library, Scopus].
- 22) \*Tiecoura Yves, Hahanov V., Alnahhal Omar, Maksimov M., Sherbin D., Yudin D. Diagnosis infrastructure of software-hardware systems // Proc. of IEEE East-West Design and Test Symposium. Sevastopol. 19-20 September 2011. P.84-89 [IEEE-Library, Scopus].
- 23) \*Hahanov V., Mishenko A., Chumachenko S., Hahanova A., Priymak A. Spam diagnosis infrastructure for individual cyberspace // Proc. of IEEE East-West Design and Test Symposium. Sevastopol. 19-20 September 2011. P.161-168 [IEEE-Library, Scopus].
- 24) \*Adamov A., Hahanov V. A security model of individual cyberspace // Proc. of IEEE East-West Design and Test Symposium. Sevastopol. 19-20 September 2011. P.169-172 [IEEE-Library, Scopus].
- 25) \*Hahanov V., Mostova K., Pashenko A. Infrastructure for testing and diagnosing multimedia devices // Proc. of IEEE East-West Design and Test Symposium. Sevastopol. 19-20 September 2011. P.394-399 [IEEE-Library, Scopus].
- 26) \*Хаханов В.И., Литвинова Е.И., Умерах Нгне Кристофер, Побеженко И.А. Архитектура логического ассоциативного мультипроцессора // Материалы 7-й Международной конференции «Автоматизация проектирования дискретных систем». 16-17 ноября 2010 (надруковано у 2011). Минск. С. 34-44.
- 27) Александров В.И., Бородина И.Б., Научный руководитель – к.т.н., асс. Зайченко С.А. Модель гетерогенного для эффективной функциональной верификации цифровых ситем // Материалы XVI Международного

- молодежного форума «Радиоэлектроника и молодежь в XXI веке». Часть 5. 17-19 апреля 2012. С.7-8.
- 28) Филиппенко И.О., Научный руководитель – д.т.н., проф. Хаханов. В.И. GPRS система слежения за подвижным объектом // Материалы XVI Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2012.часть 5. 17-19 апреля. С.13-14.
- 29) Галаган С.А., Горобец А.А., Приймак А.С. Научный руководитель – д.т.н., проф. Хаханов В.И. Модель цифровой системы для диагностирования дефектных компонентов SoC // Материалы XVI Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2012.часть 5. 17-19 апреля. С. 34-35
- 30) Мурад Али Аббас, Сайганов Георгий. Научный руководитель – д.т.н., проф. Хаханов В.И. Метод анализа матрицы активизации для поиска дефектов с заданной глубиной // Материалы XVI Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». часть 5. 17-19 апреля 2012. С.36-37.
- 31) Baghdad Ammar Avni Abbas, Филиппенко И.О. Научный руководитель – д.т.н., проф. Хаханов В.И. Синтез логических функций для встроенного диагностирования дефектов с заданной глубиной // Материалы XVI Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». Часть 5. 17-19 апреля 2012. С.38-39.
- 32) \*Nahanova Yu., Nahanov V., Litvinova E., Wajeb Gharibi. Fault detection of system level SoC model // Матеріали XI Міжнародної конференції TCSET 2012, присвяченої 60-річчю заснування радіотехнічного факультету у Львівській політехніці. 21-24 лютого 2012. Львів – Славське. С.290 [Scopus].
- 33) \*Nahanova A., Gorobets A., Priymak A., Litvinova E. Verification system for SoC HDL-code // Матеріали XI Міжнародної конференції TCSET 2012, присвяченої 60-річчю заснування радіотехнічного факультету у Львівській політехніці. 21-24 лютого 2012. Львів – Славське. С.348 [Scopus].



- 34) \*Maksimov M., Nahanov V.I., Chumachenko S.V., Baghdadi Ammar Awni Abbas. Multimatrix processor for cyberspace analysis // Матеріали XI Міжнародної конференції TCSET 2012, присвяченої 60-річчю заснування радіотехнічного факультету у Львівській політехніці. 21-24 лютого. 2012. Львів – Славське. С.243 [Scopus].
- 35) \*Dementiev S., Nahanov V., Chumachenko S., Nahanova A., Qubit models for logic circuits // Матеріали XII Міжнародної науково-технічної конференції CADSM 2013 «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці». 19-23 лютого. 2013. Львів – Поляна. С.115-119 [Scopus].
- 36) \*Nahanova I.V. Quantum process for the optimal coverage // Матеріали XII Міжнародної науково-технічної конференції CADSM 2013 «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці». 19-23 лютого 2013. Львів – Поляна. С.132-137.
- 37) \*Nahanova Y.V., Litvinova E., Nahanova I.V., Abbas B. A. A. Embedded repair of logic blocks // Матеріали XII Міжнародної науково-технічної конференції CADSM 2013 «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці». 19-23 лютого. 2013. Львів – Поляна. С.217-221 [Scopus].
- 38) Ларченко Б.Д., Шошин В.Ю., Хаханова А.В. Метод кодирования значимых компонент сегментированных изображений // Тези доповідей Четвертої Міжнародної науково-практичної конференції «Методи та засоби кодування, захисту й ущільнення інформації». 23-25 квітня 2013. Вінниця. С. 234-236.
- 39) \*Литвинова Е.И., Baghdad Ammar Awni Abbas. Модели встроенного ремонта комбинационных логических элементов // Материалы конференции "Информатика, математика, автоматика". 22-27 апреля 2013. Сумы. С. 69 [индексируются каталогом [worldcat.org](http://worldcat.org)].

- 40) \*Приймак А., Литвинова Е.И. Инфраструктура встроенного обслуживания систем на кристаллах // Материалы конференции "Информатика, математика, автоматика". 22-27 апреля 2013. Сумы. С. 70 [индексируются каталогом [worldcat.org](http://worldcat.org)].
- 41) \*Вареца В.В., Какурин Н.Я. Программное средство для анализа характеристик преобразователей кодов по методу досчета // Материалы конференции "Информатика, математика, автоматика". 22-27 апреля 2013. Сумы. С. 92 [индексируются каталогом [worldcat.org](http://worldcat.org)].
- 42) \*Вареца В.В., Макаренко А.Н. Контроль преобразования чисел в преобразователях кодов по методу накопления эквивалентов // Материалы конференции "Информатика, математика, автоматика". 22-27 апреля 2013. Сумы. С. 93 [индексируются каталогом [worldcat.org](http://worldcat.org)].
- 43) \*Какурин Н.Я., Макаренко А.Н. Повышение быстродействия преобразователей кодов Фибоначчи в двоичный код // Материалы конференции "Информатика, математика, автоматика". 22-27 апреля 2013. Сумы. С. 94 [индексируются каталогом [worldcat.org](http://worldcat.org)].
- 44) Арефьев А.А. Научный руководитель – к.т.н., доцент Зайченко С.А. Методы снижения энергопотребления цифровых систем при помощи поиска нарушений правил проектирования на RTL-уровне абстракции // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2013. часть 5. 22-24 апреля. С. 5-6.
- 45) Кокулюк А.А., Короленко А.А., Филиппенко И.О. Научный руководитель – д.т.н., проф. Кривуля Г.Ф. Программно-аппаратный интерфейс для специализированных компьютерных сетей // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». часть 5. 22-24 апреля 2013. С.13-14.
- 46) Мороз К.Э., Шестак А.С. Научный руководитель – д.т.н., проф. Хаханов В.И. Кубит-процессор оптимального покрытия (квантовый процессор) //

Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». часть 5. 22-24 апреля 2013. С.21-22.

- 47) Раков К.В., Демидчук М.А. Научный руководитель – д.т.н., проф. Хаханов В.И. Синтез куба функциональности методом суперпозиции // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». Часть 5. 22-24 апреля 2013. С.27-28.
- 48) Сарган С.В., Бовкун А.В. Научный руководитель – к.т.н., доц. Кулак Э.Н. Система обмена профессиональной информацией для удаленного использования // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2013. часть 5. 22-24 апреля. С.31-32.
- 49) Серокурова А.С., Угримова О.К. Научный руководитель – д.т.н., проф. Хаханов В.И. Модели встроенного ремонта логических блоков // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2013. часть 5. 22-24 апреля. С.33-34.
- 50) Степанова Ю.В., Меланко В.В. Научный руководитель – д.т.н., проф. Хаханов В.И. Восстановление работоспособности логических схем // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». часть 5. 22-24 апреля 2013. С.35-36.
- 51) Титаренко В.Н., Мельник А.Н. Научный руководитель – д.т.н., проф. Хаханов В.И. Имплементация куба функциональности в технологическую структуру PLD // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2013. часть 5. 22-24 апреля. С.37-38.
- 52) Цупка В.В., Лемешев А.С. Научный руководитель – д.т.н., проф. Литвинова Е.И. Инфраструктура встроенного ремонта блоков SoC // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2013. часть 5. 22-24 апреля. С.43-44.

- 53) Нестефоренко А., Хаханова Ю. Научный руководитель – д.т.н., проф. Хаханов В.И. Диагностирование вредоносных компонентов на основе таблицы функций // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2013. часть 5. 22-24 апреля. С.76-77.
- 54) Новиков И.Ю. Научный руководитель – к.т.н., доц. Филиппенко И.В. Особенности выбора ядра для решения задач цифровой обработки сигналов // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2013. часть 5. 22-24 апреля. С.78-79.
- 55) Филиппенко И.О. Научный руководитель – д.т.н., проф. Хаханов В.И. Специализированная компьютерная система «телевизионный суфлер» // Материалы XVII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». 2013. часть 5. 22-24 апреля. С.92-93.
- 56) Хаханов В.І. Чумаченко С.В., Литвинова Є.І., Філіпенко І.В., Хаханова А.В. Діагностування функціональних порушень у корпоративних мережах // Матеріали II Всеукраїнської науково-практичної конференції «Комп'ютерне моделювання та програмне забезпечення інформаційних систем і технологій». Львів. 30 мая – 1 июня 2013.С. 85-87.
- 57) Мищенко А.С., Хаханов В.И., Чумаченко С.В., Хаханова А.В., Филиппенко И.В. Инфраструктура встроенного сервисного обслуживания безопасности киберсистем // Материалы международной конференции «Наука и современность: вызовы глобализации». Киев. 25 мая 2013. С. 55-60.
- 58) Мова А.Ю, Усиченко Р.И., Бабич А.В. Исследование процессов обратной связи в видеоконференцсвязи с централизованной архитектурой // Материалы Международной научной конференции “Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI’2013)”. 20–24 мая 2013. Евпатория. С. 50-51.

- 59) Кривуля Г.Ф., Власов И.В., Павлов О.А. Оперативное функциональное диагностирование технических объектов с применением темпоральных деревьев решений // Материалы Международной научной конференции “Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI’2013)”. 20–24 мая 2013. Евпатория. С. 193-195.
- 60) \*Гузь О.А., Хаханов В.И., Литвинова Е.И., Чумаченко С.В. Модели и методы диагностирования цифровых систем на кристаллах // V Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем – 2012». Россия. Москва. 08-12 октября 2012. С. 22- 29.
- 61) \*Хаханов В.И, Murad Ali Abbas, Baghdad Ammar Avni Abbas, Хаханова И.В. Методы высокоуровневого и логического моделирования в САПР СБИС // V Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем – 2012». 08-12 октября 2012. Россия. Москва. С. 529- 534.
- 62) \*Hahanov V.I., Litvinova E.I., Frolov A., Tiecoura Yves. Models for Embedded Repairing Logic Blocks // Proc. of IEEE East-West Design and Test Symposium. IEEE. USA. Kharkov. 14-17 September 2012. P.66 – 71 [IEEE, Scopus].
- 63) \*Karyna Mostova, Oleksandr Paschenko, Vladimir Hahanov. Assertion Based Method of Functional Defects for Diagnosing and Testing Multimedia Devices // Proc. of IEEE East-West Design and Test Symposium. IEEE. USA. Kharkov. 14-17 September 2012. P.465-469 [IEEE, Scopus].
- 64) \*Volodymyr Obrizan, Alexander Adamov, Dmitry Shcherbin, Irina V. Hahanova. Transaction Level Model of Embedded Processor for Vector-Logical Analysis // Proc. of IEEE East-West Design and Test Symposium. IEEE. USA. Kharkov. 14-17 September 2012. P.497-501 [IEEE, Scopus].

- 65) \*Filippenko I.O, Maksimov M., Chugurov I., Filippenko I.V., Hahanova I.V. Models for SoC Infrastructure of Radio Frequency Identification with Code-Division Multiple // Proc. of IEEE East-West Design and Test Symposium. IEEE. USA. Kharkov. 14-17 September 2012. P.535-537 [IEEE, Scopus].
- 66) \*David Afolabi, Zhun Shen, Tomas Krilavičius, Vladimir Hahanov, Eugenia Litvinova, Svetlana Chumachenko, Ka Lok Man, Hai-Ning Liang, Nan Zhang, Tomas Žalandauskas. real-time vibration reduction in UAV's image sensors using efficient hough transform // Electronics & Electrotehnics. Литва, 2013 [Scopus].
- 67) \*Mishchenko Alexander, Dementiev Sergey, Eng Gee Lim, Zhao Wang, Gerry Juans, Ka Lok Man, Nan Zhang, Vladimir Hahanov, Eugenia Litvinova, Svetlana Chumachenko. Design and Optimization of a Planar UWB Antenna // Proc. of IEEE East-West Design and Test Symposium. Rostov-on-Don. 27-30 September 2013. P.67-71 [IEEE, Scopus].
- 68) \*Krasnorutskyi A.A., Hahanova A.V., Demedetskiy A.O. Method of Coding Bitmap Transformant to Improve Image Compression while Maintaining a Predetermined Quality Image to be Transmitted in Infocommunication Real Time Systems // Proc. of IEEE East-West Design and Test Symposium. Rostov-on-Don. 27-30 September 2013. P.461-464 [IEEE, Scopus].
- 69) \*Mova A., Usichenko R., Babich A. Diagnostic node in the RTCP feedback model for videoconferences with the centralized architecture // Матеріали XII Міжнародної науково-технічної конференції CADSM 2013 «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці». 19-23 лютого. 2013. Львів – Поляна. С.14-16 [Scopus].
- 70) \*Dementiev S., Hahanov V., Chumachenko S., Hahanova A. Qubit models for logic circuits // Матеріали XII Міжнародної науково-технічної конференції CADSM 2013 «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці». 19-23 лютого. 2013. Львів – Поляна. С.115-119 [Scopus].

- 71) \*Nahanova I.V. Quantum process for the optimal coverage // Матеріали XII Міжнародної науково-технічної конференції CADSM 2013 «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці». 19-23 лютого 2013. Львів – Поляна. С.132-137 [Scopus].
- 72) \*Nahanova Y.V., Litvinova E., Nahanova I.V., Abbas B. A. A. Embedded repair of logic blocks // Матеріали XII Міжнародної науково-технічної конференції CADSM 2013 «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці». 19-23 лютого 2013. Львів – Поляна. С.217-221 [Scopus].

#### 6.6 Договори з зацікавленими організаціями

Укладено угоду про співробітництво з Технологічним університетом Блекінге (м. Карлскрона, Швеція, 2012 р.).

Укладено угоду про співробітництво з Норвезьким університетом науки та технологій (м. Тронхейм, Норвегія, 2013 р.)

#### 6.7 Організація і проведення конференцій та семінарів

9-12 вересня 2011 р. у Севастополі (Україна) за участю студентів та аспірантів проведено 9-й IEEE EWDTs-2011: від компаній-спонсорів Aldec, Synopsys, Cadence, Kaspersky Laboratory, DataArt Lab, Tallinn Technical University. Мета симпозиуму – інтеграція технологічної культури з теоретичними розробками вчених, координація та обмін досвідом між провідними науковими школами та експертами Східної, Західної Європи та США в області моделювання і тестування цифрових систем. До участі у симпозиумі було прийнято 101 доповідь від 222 авторів з 34 країн світу (Armenia, Belarus, Brazil, Czech Republic, Côte d'Ivoire, Egypt, Estonia, Finland, Germany, India, Iran, Israel, Italy, Japan, Korea, Lebanon, Libyan Arab Jamahiriya,

Malaysia, Mexico, Nigeria, Norway, Palestine, Poland, Romania, Russia, Saudi Arabia, Singapore, Syria, Sweden, Turkey, UK, Ukraine, USA, Viet Nam). Кількість учасників – 101. Кількість пленарних та замовлених доповідей – 12. Кількість університетів та компаній – 82. Видання програми та збірки трудов симпозиуму (Proceedings) обсягом 420 стор. формату А4; за результатами проведення симпозиуму підготовлено 3 програми «Дзеркало науки» та «Європоступь» від Харківського областного телебачення (ОТБ), що популяризують досягнення вчених ХНУРЕ; включення до IEEE-Library доповідей, що були представлені на симпозиумі, серед них – 11 доповідей з ХНУРЕ

З 14 по 17 вересня 2012 р. у Харкові за ініціативою Харківського національного університету радіоелектроніки, за підтримки Академії наук прикладної радіоелектроніки, Харківського містського голови та Харківської областної держадміністрації, а також за фінансової підтримки Талліннського технологічного університету (Естонія), ІТ-компаній Aldec, Synopsys, Лабораторія Касперського, DataArt Lab пройшов ювілейний 10-й Міжнародний науковий симпозиум «IEEE East-West Design & Test Symposium – 2012». Технічним спонсором EWDTS'2012 є комп'ютерне співтовариство IEEE Computer Society Test Technology Technical Council (ТТТС). Рейтинг симпозиуму – № 6 серед прем'єр-конференцій планети з комп'ютерної інженерії. Мета симпозиуму – розширення міжнародного співробітництва та обмін досвідом між провідними вченими Західної та Східної Європи, Північної Америки та інших країн в області автоматизації проектування, тестування та верифікації електронних компонентів і систем. До участі в симпозиумі було прийнято 137 доповідей від 296 авторів із 30 країн світу (Алжир, Вірменія, Білорусь, Бразилія, Кот Дівуар, Китай, Естонія, Ефіопія, Франція, Німеччина, Індія, Іран, Італія, Лівія, Мексика, Нігерія, Палестина, Польща, Португалія, Румунія, Росія, Саудівська Аравія, Сінгапур, Південна Африка, Сирія, Швеція, Туреччина, Великобританія, Україна, США). Кількість учасників – 103. Кількість пленарних та замовних доповідей – 12. Кількість університетів і



компаній – 102. Кількість міст – 79.

З 27 по 30 вересня 2013 р. в Ростові-на-Дону за ініціативою Харківського національного університету радіоелектроніки, спільно з Донським державним технічним університетом (ДДТУ), за підтримки Академії наук прикладної радіоелектроніки, а також за фінансової підтримки Талліннського технологічного університету (Естонія), ІТ-компаній Aldec, Synopsys, Лабораторія Касперського, DataArt Lab, Agilent Technologies пройшов 11-й Міжнародний науковий симпозиум «IEEE East - West Design & Test Symposium - 2013». Технічним спонсором EWDTS'2013 є комп'ютерне співтовариство IEEE Computer Society Test Technology Technical Council (ТТТС). Рейтинг симпозиуму – № 6 серед прем'єр-конференцій планети з комп'ютерної інженерії. Мета симпозиуму – розширення міжнародного співробітництва та обмін досвідом між провідними вченими Західної та Східної Європи, Північної Америки та інших країн в області автоматизації проектування, тестування та верифікації електронних компонентів і систем. Оргкомітет симпозиуму провів велику підготовчу роботу, сповістивши понад 400 учасників та організацій України, країн СНД і далекого зарубіжжя, а також з розробки та підтримки сайту симпозиуму <http://ewdtest.com/>, доступ до якого через мережу інтернет щорічно збільшує кількість потенційних учасників. До участі у симпозиумі було прийнято 110 доповідей від 305 авторів з 23 країн світу (Алжир, Вірменія, Білорусь, Бразилія, Чехія, Кот Дівуар, Китай, Естонія, Індія, Іран, Ірак, Йорданія, Мексика, Нігерія, Норвегія, Португалія, Росія, Саудівська Аравія, Швеція, Туреччина, Україна, США, В'єтнам). Кількість учасників, які прибули на симпозиум – 84. Кількість пленарних та замовних доповідей – 7. Кількість університетів і компаній – 73. Кількість міст - 48. Представлено 81 доповідь від 94 авторів з 10 країн (Вірменія, Білорусія, Бразилія, Чехія, Індія, Іран, Португалія, Росія, Україна, США). Були видані програма і матеріали симпозиуму ( Proceedings ) обсягом 503 сторінки формату А4. EWDTS-2013 привернув увагу преси та телебачення – радіо «Ростов», офіційне видання

ДДТУ «Донський технічний», телестудія ДДТУ, програма «Дзеркало науки» Харківського Обласного телебачення, які висвітлювали роботу симпозиуму.

При кафедрі Автоматизації проектування обчислювальної техніки ХНУРЕ діє щотижневий науковий семінар «Актуальні проблеми кіберпростору», який об'єднує дослідників, викладачів, студентів та аспірантів кафедри, університету, а також зацікавлених вчених університетів Харков та України. У межах семінару обговорюються питання, пов'язані з розвитком квантових обчислень, движків кіберпростору, хмаровими обчисленнями, захистом інформації у кіберпросторі, персональним віртуальним кюберкомп'ютером, у тому числі з побудовою мультипроцесорної системи пошуку, розпізнавання та прийняття рішень для інформаційної комп'ютерної екосистеми.

#### 6.8 Виконані інноваційні проекти та проекти, що виконуються

1) Виграв конкурс проект «Curricula Development for new speciality Master of Engineering in MEMS/NEMS Design" (MastMEMS)» TEMPUS сумісно з університетом «Львівська політехніка» та Технічним університетом м. Лодзь (Польща) на 2012 – 2015 рр. – автори: д.т.н., проф. Хаханов В.І., д.т.н., проф. Чумаченко С.В. Обсяг фінансування ХНУРЕ – 153 тис. Євро.

2) Виконується проект «SEIDA» за рахунок Євросоюзу (BAITSE «Baltic Academic IT Security Exchange») сумісно зі Швецією (куратор проекту), Естонією, Латвією – автори: д.т.н., проф. Хаханов В.І., викладач, аспірант Адамов О.С., учасники – д.т.н., проф. Чумаченко С.В. Термін: 01.10.2011 – 31.05.2014. Сумісні наукові дослідження, проведення конференцій та участь у конференціях з доповідями, розробка монографії, розробка курсу лекції та лабораторного практикуму з захисту інформації, обмін студентами та викладачами з метою стажування.

## 6.9 Отримані гранти

Таблиця 6.2 – Гранти, отримані у межах НДР

Тип гранту	Кількість грантів
<p><b>На стажування/підвищення кваліфікації:</b></p> <p><b>2011 рік</b></p> <p>1) <b>1 грант</b> на стажування в Плімутському університеті (Великобританія, січень 2011 р.) – д-р техн. наук, проф. Кривуля Г.Ф. Наказ № 790К від 27.12.2010. Мета – вивчення сучасних методів проектування та діагностики комп'ютерних систем та мереж.</p> <p>2) <b>1 грант</b> на стажування в компанії Aldec (США), м. Лас-Вегас у рамках договору про співробітництво від 09.12.2009 аспіранту стаціонара 3-го року навчання Мельнику Д.В. – літо 2011. Мета – практична реалізація результатів дисертаційного дослідження. План дослідження: розробка, реалізація та тестування компонентів ALINT.</p> <p><b>2012 рік</b></p> <p>1) <b>1 грант</b> на стажування в Плімутському університеті (Великобританія, грудень 2011 р. – січень 2012 р.) – д-р техн. наук, проф. Кривуля Г.Ф. Мета – вивчення сучасних методів проектування та діагностики комп'ютерних систем та мереж.</p> <p>2) <b>1 грант</b> на стажування в Плімутському університеті (Великобританія, грудень 2012 р. – січень 2013 р.) – д-р техн. наук, проф. Кривуля Г.Ф. Мета – вивчення сучасних методів проектування та діагностики комп'ютерних систем та мереж.</p> <p>3) <b>1 грант</b> на стажування в Талліннському технологічному університеті (асп. Сорудейкин К.)</p> <p>4) <b>2 гранти</b> на магістерське стажування в університеті (Франція) – листопад 2012 – квітень 2013, згідно до договору про співробітництво між ХНУРЕ та університетом Дю Мен, відповідальний – канд. техн. наук, проф. Немченко В.П.</p> <p><b>2013 рік</b></p> <p>1) <b>3 гранти</b> на магістерське стажування в Лодзинському політехнічному університеті (Польща) – магістранти Білоус В., Скоробогатий М., Мельник М., вересень 2013.</p> <p>2) <b>3 гранти</b> на стажування викладачів у Павійському університеті (Італія) за проектом ТЕМПУС – д-р техн. наук, проф. Хаханов В.І.; д-р техн. наук, проф. Чумаченко С.В., д-р техн. наук, проф. Литвинова Є.І.</p> <p>3) <b>1 грант</b> на стажування в Плімутському університеті (Великобританія, грудень 2012 р. – січень 2013 р.) – д-р техн. наук, проф. Кривуля Г.Ф. Мета – вивчення сучасних методів проектування та діагностики комп'ютерних систем та мереж.</p>	14
<p><b>На організацію науково-навчальних заходів:</b></p> <p><b>2011 рік</b></p> <p>1) <b>3 гранти</b> на організацію та проведення IEEE EWDTs-2011 у Севастополі (Україна) 9-12 вересня 2011 р. за участю студентів та аспірантів: від компаній-спонсорів Aldec, Synopsys, Cadence, Kaspersky Laboratory, DataArt Lab, Tallinn Technical University. Цель симпозиума - интеграция технологической культуры с теоретическими разработками ученых, координация и обмен опытом между ведущими научными школами и экспертами Восточной, Западной Европы и США в области моделирования и тестирование цифровых систем. К участию в симпозиуме было принято 101 докладов от 222 авторов из 34 стран мира (Armenia, Belarus, Brazil, Czech Republic, Côte d'Ivoire, Egypt, Estonia, Finland, Germany, India, Iran, Israel, Italy, Japan, Korea, Lebanon, Libyan Arab Jamahiriya, Malaysia, Mexico, Nigeria, Norway, Palestine, Poland, Romania, Russia, Saudi Arabia, Singapore, Syria, Sweden, Turkey, UK, Ukraine, USA, Viet Nam). Количество участников – 101. Количество пленарных и заказных докладов – 12. Количество университетов и компаний – 82. Результат: видання програми та збірки трудов симпозиуму (Proceedings) обсягом 420 стор. формату А4; за результатами проведення симпозиуму підготовлено 3 програми «Дзеркало науки» та «Європоступ» від Харківського обласного телебачення (ОТБ), що популяризують досягнення вчених ХНУРЕ; включення до IEEE-Library доповідей, що були представлені на симпозиумі, серед них – 11 доповідей з ХНУРЕ.</p> <p>2) <b>1 грант</b> від Південно-Федерального університету (Ростов-на-Дону, Росія) для</p>	9

Тип гранту	Кількість грантів
<p>проведення відкритих лекцій для науковців за технологією відеоконференцій 2-4.11.2011 – д-р техн. наук, проф. Хаханов В.І. Мета – проведення відкритих наукових лекцій за тематикою НДР 259 «Кіберпростір. Движки. Сервіси», «Інженерія кіберпростору та мозкоподібні обчислення».</p> <p><b>2012 рік</b>  <b>1) 2 гранти</b> на організацію 10-го Міжнародного наукового симпозиуму «IEEE East-West Design &amp; Test Symposium - 2012», що пройшов 14 - 17 вересня 2012 у Харкові за ініціативою Харківського національного університету радіоелектроніки та за підтримкою Академії наук прикладної радіоелектроніки, а також при фінансовій підтримці Талліннського технологічного університету (Естонія), ІТ-компаній Aldec, Synopsys, Лабораторія Касперського, DataArt Lab.</p> <p><b>2013 рік</b>  <b>1) 3 гранти</b> на організацію 11-го Міжнародного наукового симпозиуму «IEEE East-West Design &amp; Test Symposium - 2013», що пройшов 27 - 30 вересня 2013 у Ростові-на-Дону за ініціативою Харківського національного університету радіоелектроніки та за підтримкою Академії наук прикладної радіоелектроніки, а також при фінансовій підтримці Талліннського технологічного університету (Естонія), ІТ-компаній Aldec, Synopsys, Лабораторія Касперського, DataArt Lab, Agilent Technologies.</p>	
<p><b>На участь у конференції/олімпіаді:</b>  <b>2011 рік:</b>  <b>1) 2 гранти</b> на участь у очному турі Міжнародної молодьожної конференції «IT Security for the Next Generation» (тур Росія та СНГ) 9-11 березня 2011, Росія, Москва, від ЗАТ «Лабораторія Касперського» – д-р техн. наук, проф. Хаханов В.І., аспірант Міщенко О.С. Мета – участь у конкурсі наукових робіт з доповіддю «Инфраструктура диагностирования СПАМА в индивидуальном кибернетическом пространстве». План дослідження: загальна характеристика дослідження (актуальність, мета, задачі), еволюція Cyber Space та Internet, інтелектуальний спам-фільтр, інфраструктура визначення спама. Результат – публікація тез доповіді на сайті Лабораторії Касперського (<a href="http://www.kasperskyacademy.com/ru/it_security_conference_archive">http://www.kasperskyacademy.com/ru/it_security_conference_archive</a>), інтернет-трансляція конкурсу, 1 місце у конкурсі, диплом, запрошення для участі у Єврофіналі конкурсу у Мюнхені, Німеччина.</p> <p><b>2) 2 гранти</b> на участь у Міжнародній молодьожній конференції «IT Security for the Next Generation» (тур Європа та Азія), Німеччина, м. Мюнхен, 12.04.2011 – 16.04.2011 від ЗАТ «Лабораторія Касперського» – д-р техн. наук, проф. Хаханов В.І., аспірант Міщенко О.С. Наказ 232К, 15А від 12.04.2011. Мета – участь у конкурсі наукових робіт з доповіддю “SPAM Diagnosis Infrastructure in Individual Cyber Space”. План дослідження: загальна характеристика дослідження (актуальність, мета, задачі), еволюція Cyber Space та Internet, інтелектуальний спам-фільтр, інфраструктура визначення спама. Результат – презентація роботи у конкурсі, виступ з доповіддю, обмін досвідом та знаннями, застосування інноваційних ідей до розв’язання проблем в області інформаційної безпеки, досвід міжнародного спілкування, подання проекту до бізнес-інкубатору, розробка та подання запиту на фінансування держбюджетної НДР «Персональний віртуальний кіберкомп’ютер та інфраструктура аналізу кіберпростору».</p> <p><b>3) 3 гранти</b> від Євросоюзу на участь в Міжнародній науковій конференції «NordSec (Nord Security)», у межах якої відбулося засідання робочої групи учасників проекту BITSE «Baltic Academic IT Security Exchange» 25-30.10.2011 – д-р техн. наук, проф. Хаханов В.І., д-р техн. наук, проф. Чумаченко С.В., викладач ЦНСІМ, аспірант каф. АПОТ Адамов О.С. Наказ № 660К від 25.10.2011. Мета – участь у конференції, координація питань щодо виконання сумісних науково-дослідних проектів. План дослідження: розробка монографії, сумісне проведення конференцій та участь у конференціях з доповідями щодо результатів досліджень. Результат – узгоджено план робіт щодо виконання проекту.</p> <p><b>2012 рік:</b>  <b>1) 1 грант</b> від організаційного комітету V Всеросійської науково-технічної конференції "Проблемы разработки перспективных микро- и наноэлектронных систем", Россию (г. Истра Московской области) для выступления с пленарным и секционным докладами – д-р техн. наук, проф. Хаханов В.І.</p> <p><b>2) 1 грант</b> від університету Бумердес (Алжир) на участь у конференції згідно з договором про співробітництво д.т.н., проф. Хаханову В.І. (3-9 червня 2012 р.)</p> <p><b>3) 1 грант</b> на участь в олімпіаді з мікроелектроніки, м. Єреван, Вірменія, жовтень 2012</p>	13

Тип гранту	Кількість грантів
(магістр гр. СКСм-12-1 Мизь В.) <b>2013 рік:</b> <b>1) 1 грант</b> на участь у нараді робочої групи проекту ВАІТSE «Baltic Academic IT Security Exchange» 22-26.04.2013, Вроцлав, Польща (ас. каф. АПОТ Адамов О.С.) <b>2) 2 гранти</b> на участь у форумі «Global Forum – 50 <sup>th</sup> Design Automation Conference», м. Остін (штат Техас), США, 02.06.2013 – 06.06.2013 (д.т.н., проф. Хаханов В.І., д.т.н., проф. Чумаченко С.В.) – не реалізовані з причин несвоєчасного отримання візи з консульства.	
<b>РАЗОМ:</b>	<b>36</b>

## 6.10 Теми захищених дисертацій

1) Асп. Ів Тієкура. «Векторно-логічна інфраструктура вбудованого тестування цифрових систем на кристалах» за спеціальністю 05.13.05 – комп'ютерні системи та компоненти (наук. керівник – д.т.н., проф. Хаханов В.І. Захист – 14.03.2012.).

2) Пошукач Філіпенко І.В. «Моделі та методи проектування систем на кристалах для радіочастотної ідентифікації з кодовим розподілом каналів» за спеціальністю 05.13.05 – комп'ютерні системи та компоненти (науковий керівник – д.т.н., проф. Хаханов В.І.). Захист – 21.03.2012.

3) Пошукач Гаркуша О.В. Моделі та методи діагностування компетентностей користувачів для забезпечення експлуатаційної готовності комп'ютерних систем за спеціальністю 05.13.05 – комп'ютерні системи та компоненти (наук. керівник – д.т.н., проф. Кривуля Г.Ф.). Захист – 18.03.2012.

4) Пошукач Герасименко К.Є. «Методи підвищення надійності обладнання керуючих систем безпеки енергоблоків АЕС за функцією захистів» за спеціальністю 05.13.05 – комп'ютерні системи та компоненти. Науковий керівник – д.т.н., проф. Хаханов В.І. Захист – 26.06.2013.

5) Мл. наук. співроб. Давидов А.А. «Автоматизація побудови та обробки продукційних правил для діагностики комп'ютерних систем» за спеціальністю 05.13.05 – комп'ютерні системи та компоненти. Науковий керівник – д.т.н., проф. Кривуля Г.Ф. Захист – 22.05.2013.

6) Рязанцев О.І. «Інформаційні технології забезпечення екологічної і техногенної безпеки промислового регіону (Лисичанськ, Рубіжне,

Севєродонецьк)» за спеціальністю 05.13.06 – інформаційні технології.  
 Науковий консультант – д-р техн. наук, проф. Кривуля Г.Ф. Захист 24.11.2012.

## 6.11 Виставки

Таблиця 6.3 – Участь у виставках

П.І.Б. розроблювачів експонату	Назва експонату	Назва виставки	Дата й місце проведення виставки	Нагороди
<b>2012</b>				
Короленко А.А., Филиппенко И.О.	Сtereo УМЗЧ с TFT-дисплеем и сенсорным вводом	16-й Международный молодежный форум	Харьков, 17-19 апреля. ХНУРЭ	
Короленко А.А.	USB звуковая аудиокарта	16-й Международный молодежный форум	Харьков, 17-19 апреля. ХНУРЭ	
Короленко А.А., Филиппенко И.О.	Отладочный стенд для изучения работы TFT-дисплеев с сенсорным вводом на базе микроконтроллера STM32	16-й Международный молодежный форум	Харьков, 17-19 апреля. ХНУРЭ	
Короленко А.А., Филиппенко И.О., Кокулюк А.А.	Концептуальное устройство для воспроизведения звуковых колебаний разночастотного спектра	16-й Международный молодежный форум	Харьков, 17-19 апреля. ХНУРЭ	
<b>2013</b>				
Короленко А.А., Филиппенко И.О.	Макет системы цифровой передачи видеосигнала	17-й Международный молодежный форум	Харьков, 22-24 апреля. ХНУРЭ	
Короленко А.А.	USB программатор микроконтроллеров AVR совместимый с AVR910	17-й Международный молодежный форум	Харьков, 22-24 апреля. ХНУРЭ	
Березин Н.	"Topological verification rules for multy-chip FPGA-devises"	Виставка у межах Міжнародної українсько-японської конференції з питань науково-промислового співробітництва	Одеса. 24-25 октября 2013	
Филиппенко И.О., Короленко А.А.	«Мишень лазерного стрелкового тренажера»	Выставка на честь випускника ХНУРЕ, конструктора космічних тренажерних систем Бориса Пільщикова.	Харьков. 2013 г.	Диплом за кращу розробку
Зиарманд А.Н.	Харьков, Пятихатки	«Создание инфраструктуры умных дорог» Международный форум «Kharkiv-IT»		III место

## 6.12 Теми захищених магістерських робіт за тематикою НДР

1. Болтонюк К.О. гр. СКСм-10-1. Спеціалізована комп'ютерна система технологічної підготовки для виробництва нестандартних компонентів цифрових виробів. Керівник магістерської роботи – проф. Литвинова Є.І. (2011)

2. Бурковський М.О. гр. СКСм-10-1. Методи та алгоритми підвищення ефективності адаптивних протоколів маршрутизації. Керівник магістерської роботи – доц. Бабіч Г.В.

3. Ганзен В.В. гр. СКСм-10-1. Дослідження методів апаратної реалізації швидкого перетворення Фур'є. Керівник магістерської роботи – проф. Хаханова І.В. (2011).

4. Горшков О.М. гр. СКСм-10-1. Дослідження методів передачі символічної та графічної інформації на віддаленний термінал. Керівник магістерської роботи – ст. викл. Обрізан В.І. (2011).

5. Дерев'янку П.Г. гр. СКСм-10-1. Проектування компонентів мозгоподібного мультипроцесора для пошуку інформації в Internet. Керівник магістерської роботи – д.т.н., проф. Хаханов В.І. (2011).

6. Добарський О.О. гр. СКСм-10-1. Програмно-апаратна модель математичного сопроцесора, що реконфігурується, для персонального комп'ютера. Керівник магістерської роботи – ст. викл. Обрізан В.І. (2011).

7. Зєнович Є. О. гр. СКСм-10-1. Моделі та методи розміщення компонентів на друкованих платах мобільних пристроїв. Керівник магістерської роботи – доц. Сиревич Є.Ю. (2011).

8. Кривонос С. С. гр. СКСм-10-1. Застосування технології багатопротокольної комутації на основі міток для побудови віртуальних мереж. Керівник магістерської роботи – доц. Шкіль О.С. (2011).

9. Мова О.Ю. гр. СКСм-10-1. Дослідження процедур підвищення ефективності передачі даних у мультисервісних мережах. Керівник магістерської роботи – доц. Бабіч Г.В. (2011).

10. Новик С. О. гр. СКСм-10-1. Апаратна верифікація цифрових пристроїв з використанням програмувальної логіки. Керівник магістерської роботи – проф. Хаханова І.В. (2011).

11. Орехова І. В. гр. СКСм-10-1. Розробка і оптимізація методів класифікації та підбору IP-Core з використанням процедур нечіткої логіки. Керівник магістерської роботи – доц. Сиревич Є.Ю. (2011).

12. Павлов О. А. гр. СКСм-10-1. Синтез керуючих кінцевих автоматів з використанням автоматного та генетичного програмування. Керівник магістерської роботи – проф. Кривуля Г.Ф. (2011).

13. Сидоренко О. В. гр. СКСм-10-1. Дослідження впливу розміру сайту на його рейтинг в пошукових системах. Керівник магістерської роботи – доц. Шкіль О.С. (2011).

14. Сідоров Д. В. гр. СКСм-10-1. Дослідження методів клітинних автоматів для розпізнавання друкарських текстів. Керівник магістерської роботи – проф. Кривуля Г.Ф. (2011).

15. Череватенко Є. О. гр. СКСм-10-1. База знань екпертної діагностичної системи з використанням нечітких мереж Петрі. Керівник магістерської роботи – проф. Кривуля Г.Ф. (2011).

16. Александров В. І. гр. СКСм-11-1. Методи ранжування тестів моделей цифрових систем за гетерогенними метриками функціонального покриття. Керівник магістерської роботи – доц. Зайченко С.О. (2012).

17. Бородіна І. Б. гр. СКСм-11-1. Модель гетерогенного покриття для ефективної функціональної верифікації цифрових систем. Керівник магістерської роботи – доц. Зайченко С.О. (2012).



18. Голіней С. В. гр. СКСм-11-1. Спеціалізований програмно-апаратний обчислювальний комплекс з розподіленою архітектурою. Керівник магістерської роботи – проф. Кривуля Г.Ф. (2012).

19. Іваненко В. В. СКСм-11-1. Тестування HDL-моделей із застосуванням сучасних технологій верифікації. Керівник магістерської роботи – проф. Хаханова І.В. (2012).

20. Педич М.М. гр. СКСм-11-1. Комплексна тема. Методи та спеціалізовані комп'ютерні засоби для телеметричного збору та обліку технологічних даних. Технічне забезпечення. Керівник магістерської роботи – проф. Кривуля Г.Ф. (2012).

21. Шепель С. В. гр. СКСм-11-1. Комплексна тема. Методи та спеціалізовані комп'ютерні засоби для телеметричного збору та обліку технологічних даних. Програмне забезпечення. Керівник магістерської роботи – проф. Кривуля Г.Ф. (2012).

22. Шевченко В. В. гр. СКСм-11-1. Аналіз функціонування HDL-моделей із застосуванням сучасних технологій верифікації. Керівник магістерської роботи – проф. Хаханова І.В. (2012).

23. Ареф`єв А. О. гр. СКСм-12-1. Методи зменшення енергоспоживання цифрових систем на базі пошуку зразків порушень правил проектування на RTL-рівні абстракції. Керівник магістерської роботи – проф. Чумаченко С.В. (2013).

24. Короленко О. О. гр. СКСм-12-1. Розробка спеціалізованої комп'ютерної моделі для вивчення впливу шкідливих чинників на виробництві. Керівник магістерської роботи – проф. Хаханова І.В. (2013).

25. Лобода А. В. гр. СКСм-12-1. Методи обробки мережевого потокового відео з використанням спеціалізованих бібліотек мови програмування Java. Керівник магістерської роботи – доц. Шкіль О.С. (2013).

26. Мізь В. О. гр. СКСм-12-1. Розробка спеціалізованої комп'ютерної системи стеження з використанням мобільних технологій. Керівник магістерської роботи – проф. Литвинова Є.І.

27. Хаханова Ю.В. гр. СКСм-12-1. Моделі та методи інфраструктури вбудованого ремонту систем на кристалах. Керівник магістерської роботи – проф. Литвинова Є.І. (2013).

28. Шишов М. М. гр. СКСм-12-1. Розробка спеціалізованої комп'ютерної системи навігації з використанням мобільних технологій. Керівник магістерської роботи – проф. Литвинова Є.І. (2013).

### 6.13 Залучення позабюджетних коштів

Господоговори виконані та, що виконуються (назва, замовник, загальна сума, доля, що виконана власними силами конкретні результати отримані по договору). Загальний обсяг позабюджетних робіт у 2011-2013 рр. – 30000 грн.

Таблиця 6.4 – Залучення позабюджетних коштів

№ п/п	Назва	Замовник	Загальна сума	Доля, що виконана власними силами	Конкретні результати, отримані по договору
1	Господоговірна НДР 11-01 «Розробка концептуальної моделі процесу верифікації проектів цифрових систем, що враховують обмеження в форматі UPF щодо ефективного споживання електричної енергії». Договір № 11-01 від 04.01.11 Науковий керівник – д-р техн. наук, проф. Чумаченко С.В.	ТОВ «Алдек-КТС»	Обсяг фінансування: 15000 грив.	100%	Згідно з ТЗ: модель процесу сумісного імітаційного моделювання та верифікації HDL-проектів разом з UPF-обмеженнями. Висування пропозицій щодо способів уточнення UPF-моделей з використанням статистичних даних переключення керуючих сигналів. Оцінка обчислювальної складності запропонованого процесу верифікації.
2	Господоговірна НДР про надання консультаційних послуг	Сєверодонецьке НВО «Імпульс»	Обсяг фінансування: 15000 грив.	100%	Надання консультаційних послуг

## 6.14 Розробки

Таблиця 6.5 – Розробки за тематикою виконаної НДР

№ з/п	Назва та автори розробки	Показники результативності, переваги над аналогами, економічний, соціальний ефект	Місце впровадження (назва організації, відомча належність, адреса)	Дата впровадження	Результати, які отримано ХНУРЕ від впровадження
<b>2011</b>					
1	Векторно-логічна інфраструктура вбудованого тесування цифрових систем на кристалах. Автор: асп. Ів Тікура, д.т.н., проф. Хаханов В.І.	Програмно-апаратний комплекс у вигляді інфраструктури тестування та верифікації, що дає можливість суттєво (20%) підвищити тестопригодність цифрової системи на кристали	ТОВ «Алдек-Харків»	Довідка про впровадження у науково-дослідну та виробничу діяльність компанії ООО «Алдек» №14 від 10.10.2011 згідно з Договором №3 від 09.12.2009 про співробітництво між ХНУРЕ та корпорацією Aldec Inc (США)	Програмно-апаратний комплекс у вигляді інфраструктури тестування та верифікації
2	Захист індивідуального кібернетичного простору (інфраструктура діагностування спаму у індивідуальному кібернетичному просторі) д.т.н., проф. каф. АПОТ, декан факультету КІУ Хаханов В.І., д.т.н., проф. каф. АПОТ Чумаченко С.В., асп. каф. АПОТ Міщенко О.С., студенти гр. КІ-08-6 Зацарний А.В., Хаханова Ю.В.	Очікувані результати та їх ринкова привабливість полягають у наступному: 1) Інфраструктура вбудованого захисту програмного коду від несанкціонованої модифікації, що призводить до зміни функціональності. 2) Надмірність інфраструктури програмного коду, яка автоматично синтезується на стадії проектування та верифікації, становить не більше 5% від специфікованої функціональності. 3) Ринкова привабливість інфраструктури, обумовлена різноманіттям програмних	ЗАТ «Лабораторія Касперського», Москва, Росія	Захист проекту на конкурсі наукових розробок від Лабораторії Касперського 9-11.03.2011, Москва, Росія, Лабораторія Касперського; 14-15.04.2011 Мюнхен, Лабораторія Касперського	Прототип інфраструктури захисту індивідуального кіберпростору. Диплом за 1 місце у конкурсі наукових проектів від Лабораторії Касперського.

№ з/п	Назва та автори розробки	Показники результативності, переваги над аналогами, економічний, соціальний ефект	Місце впровадження (назва організації, відомча належність, адреса)	Дата впровадження	Результати, які отримано ХНУРЕ від впровадження
		<p>продуктів, помноженої на рівень продаж кожного виробу, складає в рік близько одного мільярда примірників. 4) Вартість створення інфраструктури для програмного продукту становить 20% витрат від розробки функціонального коду. Якщо рівень продаж – не менше 500 копій, то витрати на створення вбудованого антивіруса цілком можуть бути окупаємі протягом двох років.</p>			
3	<p>Концептуальна модель процесу верифікації проектів цифрових систем, що враховують обмеження в форматі UPF щодо ефективного споживання електричної енергії. Автор: с.н.с. Таранов В.Б., д.т.н., проф. Чумаченко С.В.</p>	<p>Модель процесу сумісного імітаційного моделювання та верифікації HDL-проектів разом з UPF-обмеженнями, що містить пропозиції щодо способів уточнення UPF-моделей з використанням статистичних даних переключення керуючих сигналів та оцінку обчислювальної складності запропонованого процесу верифікації.</p>	ТОВ «Алдек-Харків»	<p>Акт здачі-приймання науково-технічної продукції від 30.04.2011 р. за договором № 11-01 від 04.01.2011</p>	<p>Модель процесу сумісного імітаційного моделювання та верифікації HDL-проектів разом з UPF-обмеженнями.</p>
<b>2012</b>					
4	<p>Система радіочастотної ідентифікації</p>	<p>Розроблені моделі та методи проектування інфраструктури радіочастотної ідентифікації на стадіях проектування дозволяють</p>	ТОВ «РадіоТрейс»	<p>Довідка про впровадження у науково-дослідну та виробничу діяльність ТОВ «Радіо Трейс» від 25.01.2011</p>	<p>SoC орієнтована інфраструктура радіочастотної ідентифікації з кодовим розподіленням каналів</p>

№ з/п	Назва та автори розробки	Показники результативності, переваги над аналогами, економічний, соціальний ефект	Місце впровадження (назва організації, відомча належність, адреса)	Дата впровадження	Результати, які отримано ХНУРЕ від впровадження
		підвищити ефективність створення нових систем, що відповідають вимогам замовника. Забезпечення збереження об'єктів за рахунок міток радіочастотної ідентифікації			
5	Експертна діагностична система	Забезпечення стабільної роботи кластера за умови максимальної завантаження, що є критично важливим для виконання проектних графіків розробок. Своєчасне виявлення та діагностика несправних вузлів комп'ютерної техніки в результаті використання експертної системи суттєво зменшує ризики, що пов'язані з затримками при виконанні проектних задач.	Сєверодонецьке НВО «Імпульс»	Довідка про впровадження у науково-дослідну та виробничу діяльність Сєверодонецьке НВО «Імпульс»	Експертна діагностична система для діагностики технічного стану обладнання спеціалізованого комп'ютерного кластеру загальною обчислювальною міцністю 32 ПК різної конфігурації та термінів експлуатації
<b>2013</b>					
6	Методи функціонального тестування критичних систем керування	Розроблено математичний апарат методу підвищення контролепридатності критичних систем управління, відмінною рисою якого є побудова логічних елементів алгоритмів захисту без використання традиційної бінарної арифметики за рахунок застосування багатомірного (багаторозрядного) зображення вхідних, вихідних даних та процесів їх обробки	ЧАО Сєверодонецьке НВО «Імпульс»	Акт про впровадження 12.03.2012	Програмно-апаратні засоби тестування цифрового обладнання захисту, побудованого з використанням функціональних елементів з арифметичними операціями, що використовуються при проведенні лабораторних робіт, пов'язаних з синтезом цифрових систем на кристалах, що імплементується у програмовану

№ з/п	Назва та автори розробки	Показники результативності, переваги над аналогами, економічний, соціальний ефект	Місце впровадження (назва організації, відомча належність, адреса)	Дата впровадження	Результати, які отримано ХНУРЕ від впровадження
					логіку, у процесі вивчення курсів «Проектування цифрових систем на ПЛІС», «Основи автоматизації проектування».
7	Методи пошуку помилок проектування в моделях цифрових пристроїв	Розроблена методика проведення діагностичного експерименту при верифікації HDL-коду, що дозволяє автоматично застосовувати алгоритми пошуку дефектів в залежності від форми специфікації і стилю написання HDL-моделі. Запропонований метод зворотного відстеження для графової моделі HDL-коду дозволив значно підвищити глибину пошуку помилок проектування (до операторів HDL-коду) в умовах неповної специфікації на проєктований пристрій і значно скоротити загальний час верифікації HDL-моделі. Застосування розробленої методики дозволило на 20-30% скоротити час на пошук помилок проектування HDL-моделей і підвищити глибину пошуку до помилкового HDL-оператора.	ООО «Альдек-КТС»	Довідка про впровадження № 32/13 від 15.05.2013	Використання зазначених методів у системах верифікації HDL-моделей при автоматизованому проєктуванні цифрових пристроїв, які подані на мовах опису апаратури.

### 6.15 Використання результатів НДР у навчальному процесі

Програмно-апаратні засоби тестування цифрового обладнання захисту, побудованого з використанням функціональних елементів з арифметичними операціями, використовуються при проведенні лабораторних робіт, пов'язаних з синтезом цифрових систем на кристалах, що імплементується у програмовану логіку, у процесі вивчення курсів «Проектування цифрових систем на ПЛІС», «Основи автоматизації проектування».

Використання розроблених методів у системах верифікації HDL-моделей при автоматизованому проектуванні цифрових пристроїв, які подані на мовах опису апаратури, використовується у навчальному процесі кафедри АПОТ при викладанні навчальних дисциплін: 1) «Логічне моделювання» для бакалаврів напрямку «Комп'ютерна інженерія» у лекційному матеріалі по темі «Верифікація моделей цифрових пристроїв» та у лабораторних роботах по темі «Побудова мовних моделей цифрових пристроїв». 2) «Основи комп'ютерної діагностики» для бакалаврів напрямку «Комп'ютерна інженерія» у лекційному матеріалі та практичних заняттях по темі «Методи пошуку дефектів у цифрових пристроях». 3) «Теорія проектування СКС» для магістрів спеціальності «Спеціалізовані комп'ютерні системи» у лекційному матеріалі по темі «Системи автоматизованого проектування цифрових пристроїв».

Результати дослідження включено до навчального посібника з курсу «Дискретна математика» для студентів напряму «комп'ютерна інженерія» (додаткові розділи, тема «Векторно-логічна алгебра»).

## СПИСОК ПОСИЛАНЬ

1. Singer G. The Challenges of Nanotechnology and Gigacomplexity / G. Singer, R. Galivanche, S. Patil, M. Tripp // IEEE Design & Test of Computers.– Jan./Feb., 2009.– P. 88-92.
2. Шагурин И. Системы на кристалле. Особенности реализации и перспективы применения / И. Шагурин // Электронные компоненты. – 2009.– №1.– С. 37-39.
3. Zorian Yervant. Tutorial on EWDTS 2007. Embedding Infrastructure IP for SOC Yield Improvement / Yervant Zorian // Proceedings of the 39th Design Automation Conference.– New Orleans, LA, USA.– 2002.– P. 709-712.
4. Zorian Yervant. Test Strategies for System-in-Package / Yervant Zorian // The Plenary Paper of IEEE East-West Design & Test Symposium (EWDTS'08).– Lviv, Ukraine.– October 9-12, 2008.
5. Bhattacharya N. SoftwareHardware Hybrid Systems Verification / N. Bhattacharya // Software Testing, Verification and Validation (ICST).– 2011.– P.435-438.
6. Zilic Z. Challenges of Rapidly Emerging Consumer Space Multiprocessors / Z. Zilic, P. Mishra, S. Shukla // Design & Test of Computers.– May-June 2011.– Vol. 28, No. 3.– P.52-53.
7. Chakrabarty K. Embedded memory technologies: Present and future / Krishnendu Chakrabarty // Design & Test of Computers.– Jan.-Feb. 2011.–Vol.28, No.1.– P.4.
8. Kim C. H. Guest editors' introduction: Nanoscale Memories Pose Unique Challenges / Chris H. Kim; Leland Chang // Design & Test of Computers.– Jan.-Feb. 2011.– Vol.28, No.1.– P.6-8.
9. Itoh K. Embedded Memories: Progress and a Look into the Future / K. Itoh // Design & Test of Computers.– Jan.-Feb. 2011.– Vol. 28, No.1.– P.10-13.



10. Kularatna N. Electronic circuit design from concept to implementation / N. Kularatna.– CRC press NW. – 2008.– 504 c.
11. Chakrabarty K. Toward Bug-free Multicore SoC Architectures: System Validation with Transaction-Level Models / K. Chakrabarty // IEEE Design & Test of Computers.– 2011.– Vol. 28, Issue 3.– P. 4.
12. Mishra P. Guest Editors' Introduction:Multicore SoC Validation withTransaction-Level Models / Prabhat Mishra, Sandeep Shukla, Zeljko Zilic // IEEE Design & Test of Computers.– 2011.– Vol. 28, Issue 3.– P. 6-7.
13. Wagner I. Reversi: Post-silicon validation system for modern microprocessors / I. Wagner, V. Bertacco // Computer Design. ICCD.– Oct. 2008.– P.307-314.
14. Brown S. TLM-Driven Design and Verification – Time For A Methodology Shift. White paper / Steve Brown // Cadence Design Systems, Inc. [http://www.cadence.com/rl/Resources/white\\_papers/tlm-wp.pdf](http://www.cadence.com/rl/Resources/white_papers/tlm-wp.pdf)
15. Abdi S. Automatic TLM Generation for Early Validation of Multicore Systems / S. Abdi; G. Schirner; Yonghyun Hwang; D.D.Gajski; Yu Lochi // IEEE Design & Test of Computers.– 2011.– Vol. 28, Issue 3.– P. 10-19.
16. Dongwan S. Automatic generation of transaction level models for rapid design space exploration / Shin Dongwan, A. Gerstlauer, Peng Junyu, R. Domer, D.D. Gajski // Hardware/Software Codesign and System Synthesis. CODES+ISSS '06.– Oct. 2006.– P.64-69.
17. Coussy P. An Introduction to High-Level Synthesis / P. Coussy, D.D. Gajski, M. Meredith, A. Takach // Design & Test of Computers.– July-Aug. 2009.– Vol.26, No.4.– P.8-17.
18. Gajski D.D. Embedded System Design: Modeling, Synthesis and Verification / D.D. Gajski et al.– Springer.– 2009.– 358 p.
19. Densmore D. A Platform-Based taxonomy for ESL Design / D. Densmore, R. Passerone, A. Sangiovanni-Vincentelli // Design&Test of computers.– September-October, 2006.– P. 359-373.

20. Hwang Y. Cycle-approximate Retargetable Performance Estimation at the Transaction / Y. Hwang, S. Abdi, D. Gajski // Design, Automation and Test in Europe, DATE08.– 2008.– P. 3-8.

21. Strum M. Comparing two testbench methods for hierarchical functional verification of a bluetooth baseband adaptor / M. Strum, Wang Jiang Chau, E.L. Romero // Hardware/Software Codesign and System Synthesis, CODES+ISSS '05.– 2005.– PP. 327 – 332.

22. Bodean D. G. Constrained-random test bench for synthesis: Technique, tools and results / D. G. Bodean, Gh. C. Bodean // Automation Quality and Testing Robotics (AQTR).– May 2010.– Vol. 2.– P.1-6.

23. Di Guglielmo G. Semi-formal functional verification by EFSM traversing via NuSMV / G. Di Guglielmo, F. Fummi, G. Pravadelli, S. Soffia, M. Roveri // High Level Design Validation and Test Workshop (HLDVT).– June 2010.–P. 58-65.

24. Bawadekji M. TLM protocol compliance checking at the Electronic System Level / Mohamed Bawadekji, Daniel Grosse, Rolf Drechsler // Design and Diagnostics of Electronic Circuits & Systems (DDECS).– April 2011.– P. 435-440.

25. Le H.M. Towards analyzing functional coverage in SystemC TLM property checking / H.M. Le, D. Grosse, R. Drechsler // High Level Design Validation and Test Workshop (HLDVT).– June 2010.– P. 67-74.

26. Charles L. Advanced Techniques for Building Robust Testbenches with DesignWare Verification IP and Verification Methodology Manual (VMM) for SystemVerilog / Li Charles, Doshi Ashesh.– 2006.– Synopsys, Inc.– 16 p.

27. Charles L. Five Vital Steps to a Robust Testbench with DesignWare Verification IP and Verification Methodology Manual (VMM) for SystemVerilog/ Li Charles, Doshi Ashesh.– 2006.– Synopsys, Inc.– 17 p.

28. Sungchul P. A Functional Coverage Metric for Estimating the Gate-Level Fault Coverage of Functional Tests / Park Sungchul, Chen Li, K. Parvathala Praveen, Patil Srinivas, Pomeranz Irith // Test Conference. ITC '06. – Oct. 2006.–P. 1-10.

29. IEEE Standard for System Verilog-Unified Hardware Design, Specification, and Verification Language.– IEEE STD 1800-2009.– 2009.– 1285 p.
30. Miczo A. Digital Logic Testing and Simulation. 2nd ed. / Alexander Miczo.– Wiley New Jersey. –2003.– 673 c.
31. Wang L.-T. System-on-chip test architectures: nanometer design for testability/ Laung-Terng Wang, Charles Stroud, Nur Touba // Morgan Kaufmann. – Elsevier. – 2008.
32. Guzey O. Coverage-directed test generation through automatic constraint extraction / O. Guzey, L.-C. Wang // High Level Design Validation and Test Workshop. HLVDT. – Nov. 2007.– P.151-158.
33. Kitchen N. Stimulus generation for constrained random simulation / N. Kitchen, A. Kuehlmann // Computer-Aided Design. ICCAD. – Nov. 2007.– P. 258-265.
34. Rahagude N. DFT + DFD: An Integrated Method for Design for Testability and Diagnosability / N. Rahagude, M. Chandrasekar, M.S. Hsiao // Test Symposium (ATS). – Dec. 2010.– P. 218-223.
35. Sanyal A. RT-level design-for-testability and expansion of functional test sequences for enhanced defect coverage / A. Sanyal, K. Chakrabarty, M. Yilmaz, H. Fujiwara // Test Conference (ITC). – Nov. 2010.– P. 1-10.
36. Lechuga Y. Design-for-Test method for high-speed ADCs: Behavioral description and optimization / Y. Lechuga, R. Mozuelos, M. Martinez, S. Bracho // Design and Diagnostics of Electronic Circuits & Systems (DDECS). – April 2011.– P. 35-40.
37. Kahng A.B. Design for manufacturability: Then and now / A.B. Kahng // Design & Test of Computers.– Jan.-Feb. 2011.– Vol. 28, No.1.– P.76-77.
38. <http://www.vsi.org>
39. IEEE 1500 Web Site. <http://grouper.ieee.org/groups/1500/>.
40. IEEE P1500 Standard for Embedded Core Test (SECT).– 2005.

41. Marinissen E.J. IEEE Std. 1500 Introduction / Erik Jan Marinissen, Rohit Kapur, Yervant Zorian // Proceedings on International Test Conference.– 2000.– P.770 -777.
42. Krolikoski S. Evolution of EDA standards worldwide / S. Krolikoski // Design & Test of Computers.– Jan.-Feb. 2011.– Vol. 28, No.1.– P. 72-75.
43. Marinissen Erik Jan. IEEE Std 1500 Enables Modular SoC Testing / Erik Jan Marinissen, Yervant Zorian // Design & Test of Computers.– January/February 2009. – P.8-16.
44. Benso A. Automating the IEEE std. 1500 compliance verification for embedded cores / A. Benso, S. Di Carlo, P. Prinetto, A. Bosio // High Level Design Validation and Test Workshop. HLVD. – Nov. 2007.– P.171-178.
45. Nadeau-Dostie B. Improved Core Isolation and Access for Hierarchical Embedded Test / B. Nadeau-Dostie, S.M.I. Adham, R. Abbott // Design & Test of Computers. – Jan.-Feb. 2009.– Vol. 26, No.1.– P. 18-25.
46. Wang L.-T. Turbo1500: Core-Based Design for Test and Diagnosis / Laung-Terng Wang and others // Design & Test of Computers.– Jan.-Feb. 2009.– Vol. 26, No.1.– P.26-35.
47. Chakravadhanula K. Automating IEEE 1500 Core Test–An EDA Perspective / K. Chakravadhanula, V. Chickermane // Design & Test of Computers.– May-June 2009.– Vol.26, No.3.– P.6-15.
48. Noia B. Test-wrapper optimization for embedded cores in TSV-based three-dimensional SOCs / B. Noia, K. Chakrabarty, Xie Yuan // Computer Design. ICCD. – Oct. 2009.– P.70-77.
49. Dongsoo L. Memory-based embedded digital ATE / Lee Dongsoo, Park Sang Phill, Goel Ashish, Roy Kaushik // 29th VLSI Test Symposium (VTS), 2011.– 2011.– P. 266-271.
50. Abramovici M. BIST-Based Delay-Fault Testing in FPGAs / Miron Abramovici, Charles E. Stroud // Journal of Electronic Testing: Theory & Applications.– 2003.– Vol. 19, No. 5.– P. 549-558.

51. Ulbricht M. A new hierarchical built-in self-test with on-chip diagnosis for VLIW processors / Markus Ulbricht, Mario Scholzel, Tobias Koal, Heinrich Theodor Vierhaus // Design and Diagnostics of Electronic Circuits & Systems (DDECS). – April 2011.– P.143-146.

52. Elm M. BISD: Scan-based Built-In self-diagnosis / M. Elm, H.-J. Wunderlich // Design, Automation & Test in Europe Conference & Exhibition (DATE).– March 2010.– P. 1243-1248.

53. Huang Yu-Jen. A built-in self-test scheme for the post-bond test of TSVs in 3D ICs / Yu-Jen Huang, Jin-Fu Li, Ji-Jan Chen, Ding-Ming Kwai, Yung-Fa Chou, Cheng-Wen Wu // VLSI Test Symposium (VTS).– May 2011.– P.20-25.

54. Shianling Wu. Logic BIST Architecture Using Staggered Launch-on-Shift for Testing Designs Containing Asynchronous Clock Domains / Wu Shianling, Laung-Terng Wang, Yu Lizhen, H. Furukawa, Wen Xiaoqing, W.-B. Jone, N.A. Touba, Zhao Feifei, Liu Jinsong, Hao-Jan Chao, Li Fangfang, Jiang Zhigang // Defect and Fault Tolerance in VLSI Systems (DFT). – Oct. 2010.– P. 358-366.

55. Tsu-Wei Tseng. A Shared Parallel Built-In Self-Repair Scheme for Random Access Memories in SOCs / Tsu-Wei Tseng, Jin-Fu Li // Test Conference. ITC 2008. – Oct. 2008.– P.1-9.

56. Novak O. Handbook of testing electronic systems / O. Novak, E. Gramatova, R.Ubar.– Czech University Publishing House.– 2005.– 402 p.

57. Brown S.D. Fundamentals of digital logic with VHDL design / S.D. Brown, Z.G. Vranesic.– USA: McGraw-Hill Companies. – 2000. – 840p. ([www.mhhe.com/brawnvransic](http://www.mhhe.com/brawnvransic)).

58. Bennetts Ben. DFT course / Ben Bennetts // <http://www.ben@dft.co.uk>

59. Math resources: <http://www.math.com/>

60. Горбатов В.А. Основы дискретной математики / В.А. Горбатов.– М.: Высшая школа. – 1986. – 311 с.

61. Smith L. 3D Packaging Applications, Requirements, Infrastructure and Technologies / L. Smith // Fourth Annual International Wafer-Level Packaging Conference. – San Jose, California. – Sept., 2007.– P. 134-140.

62. Какурин Н.Я. Регистр сдвига / Н.Я. Какурин, В.И. Хаханов, В.Г. Лобода, А.Н. Какурина.– А.С. №1439682. 22.07.88.– 4с.

63. Бондаренко М.Ф. О мозгоподобных ЭВМ / М.Ф. Бондаренко, З.В. Дударь, И.А. Ефимова, В.А. Лещинский, С.Ю. Шабанов–Кушнаренко // Радиоэлектроника и информатика.– Харьков: ХНУРЭ.–2004.–№ 2.–С. 89-105.

64. Бондаренко М.Ф. Теория интеллекта / М.Ф. Бондаренко, Ю.П. Шабанов–Кушнаренко.– Учебник.– Харьков: СМИТ.– 2006.– 592 с.

65. Акритас А. Основы компьютерной алгебры с приложениями: Пер. с англ. / А. Акритас.– М.: Мир.– 1994. – 544 с.

66. Аттетков А.В. Методы оптимизации / А.В. Аттетков, С.В. Галкин, В.С. Зарубин.– М.: Издательство МГТУ им. Н.Э. Баумана.– 2003.– 440 с.

67. Проектирование и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. – Харьков: ХНУРЭ.– 2009.– 484с.

68. Хаханов В.И., Хаханова И.В., Литвинова Е.И., Гузь О.А. Проектирование и верификация цифровых систем на кристаллах. Verilog & System Verilog: Харьков.– Новое слово.– 2010.– 528с.

69. Cohen A.A. Brain-like computer architecture / IEEE International Workshop on Publication Date.– 12-16 May, 2003.– P. 12-23. 15. Cohen A.A. Addressing architecture for Brain-like Massively Parallel Computers / Euromicro Symposium on Digital System Design (DSD'04).– 2004.– P. 594-597.

70. Основы технической диагностики / Под. ред. П.П.Пархоменко.– М.: Энергия.– 1976.– 460с.

71. Пархоменко П.П. Основы технической диагностики (Оптимизация алгоритмов диагностирования, аппаратурные средства) / П.П. Пархоменко, Е.С. Согомонян. Под ред. П.П. Пархоменко.– М.: Энергия.– 1981.– 320 с.

72. Инфраструктура мозгоподобных вычислительных процессов / М.Ф. Бондаренко, О.А. Гузь, В.И. Хаханов, Ю.П. Шабанов-Кушнарченко.– Харьков: Новое Слово.– 2010.– 160 с.

73. Семенец В.В. Проектирование цифровых систем с использованием языка VHDL / В.В. Семенец, И.В. Хаханова, В.И. Хаханов.– Харьков: ХНУРЭ.– 2003.– 492 с.

74. Хаханов В.И. VHDL+Verilog = синтез за минуты / В.И. Хаханов, И.В. Хаханова.– Харьков: ХНУРЭ.– 2006.– 264с.

75. Хаханов В.И. Техническая диагностика цифровых и микропроцессорных структур: Учебник / В.И. Хаханов.– К.: ИСИО.– 1995.– 242с.

76. Скобцов Ю.А. Логическое моделирование и тестирование цифровых устройств / Ю.А. Скобцов, В.Ю. Скобцов.– Донецк: ИПММ НАН Украины, ДонНТУ.– 2005.– 436 с.

77. IEEE Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture IEEE Std 1149.7-2009. – 985 p.

78. Da Silva F. The Core Test Wrapper Handbook. Rationale and Application of IEEE Std. 1500™ / F. Da Silva, T. McLaurin, T. Waayers.– Springer.– 2006.– XXIX.– 276 p.

79. Marinissen E.J. Guest Editors' Introduction: The Status of IEEE Std 1500 / E.J. Marinissen, Yervant Zorian // IEEE Design & Test of Computers.– 2009.– No26(1).– P.6-7.

80. Marinissen E.J. Testing TSV-based three-dimensional stacked ICs / E.J. Marinissen // DATE 2010.– 2010.– P.1689-1694.

81. Ubar R. Embedded diagnosis in digital systems / R. Ubar, S. Kostin, J. Raik // 26th International Conference “Microelectronics”, MIEL 2008. – 2008.– P. 421-424.

82. Elm M. Scan Chain Organization for Embedded Diagnosis / M. Elm, H.-J. Wunderlich // Design, Automation and Test in Europe, DATE '08.– 2008.– P. 468–473.

83. Dervisoglu Bulent I. A Unified DFT Architecture for Use with IEEE 1149.1 and VSIA/IEEE P1500 Compliant Test Access Controllers / Bulent I. Dervisoglu // Proceedings of the Design Automation Conference.–2001.– P. 53-58.

84. Chenlong Hu. Hardware design and realization of matrix converter based on DSP & CPLD / Hu Chenlong, Yang Ping, Xiao Ying, Zhou Shaoxiong // 3rd International Conf. Power Electronics Systems and Applications.–2009.–P. 1-5.

85. Dave N. Hardware Acceleration of Matrix Multiplication on a Xilinx FPGA / N. Dave, K. Fleming, King Myron, M. Pellauer, M. Vijayaraghavan // 5th IEEE/ACM International Conference Formal Methods and Models for Codesign.– 2007.– P.97-100.

86. Loucks W.M. A Vector Processor Based on One-Bit Microprocessors / W.M. Loucks, M. Snelgrove, S.G. Zaky // IEEE Micro.–Volume 2, Issue 1.– 1982.– P. 53-62.

87. Hilewitz Y. Bit matrix multiplication in commodity processors / Y. Hilewitz, C. Lauradoux, R.B. Lee // International Conference Application-Specific Systems, Architectures and Processors.– 2008.– P. 7-12.

88. Кузнецов О.П. Быстрые процессы мозга и обработка образов / О.П. Кузнецов // Новости искусственного интеллекта.– 1998.– №2.

89. Васильев С.Н. Интеллектуальное управление динамическими системами / С.Н. Васильев, А.К. Жерлов, Е.А. Федосов, Б.Е. Федунов.– М.: Физико-математическая литература.– 2000.– 352 с.

90. Липаев В.В. Программная инженерия. Методологические основы. Учебник / В.В. Липаев.– Москва: Теис.– 2006.– 608 с.

91. Гайдук С.М. Сферический мультипроцессор PRUS для решения булевых уравнений / С.М. Гайдук, В.И. Хаханов, В.И. Обризан, Е.А. Каменюка // Радиоэлектроника и информатика. Харьков. 2004. № 4(29). С.107-116.



92. Аттетков А.В. Методы оптимизации / А.В. Аттетков, С.В. Галкин, В.С. Зарубин.– Москва: Издательство МГТУ им. Н.Э. Баумана.– 2003.– 440 с.
93. Abramovici M. Digital System Testing and Testable Design / M. Abramovici, M.A. Breuer and A.D. Friedman.– Comp. Sc. Press.– 1998.– 652 p.
94. Densmore D. A Platform–Based taxonomy for ESL Design / Douglas Densmore, Roberto Passerone, Alberto Sangiovanni–Vincentelli // Design & Test of computers.– 2006.– P. 359–373.
95. Автоматизация диагностирования электронных устройств/ Ю.В.Малышенко и др./ Под ред. В.П.Чипулиса.– М.: Энергоатомиздат, 1986.– 216с.
96. Трахтенгерц Э.А. Компьютерные методы реализации экономических и информационных управленческих решений / Э.А. Трахтенгерц. – СИНТЕГ.– 2009.– 396 с.
97. Mizumoto K. A multi matrix-processor core architecture for real-time image processing SoC / K. Mizumoto, T. Tanizaki, S. Kobayashi, M. Nakajima, T. Gyohten, H. Yamasaki, H. Noda, M. Higashida, Y. Okuno, K. Arimoto // Solid-State Circuits Conference. ASSCC '07.– 2007.– P. 180 – 183.
98. Hirokatsu S. Design of a Processing Element Based on Quaternary Differential Logic for a Multi-Core SIMD Processor / Hirokatsu Shirahama, Akira Mochizuki, Takahiro Hanyu, Masami Nakajima and Kazutami Arimoto // Research Institute of Electrical Communication.– Tohoku University.– 2007.– P. 80 – 85.
99. Yarom I. Smart-Lint: Improving the Verification Flow / I. Yarom, V. Patil // Proceedings of Haifa Verification Conference. – 2006. – P. 81-91.
100. Stadler M. Functional verification of intellectual properties (IP): a simulation based solution for an application-specific instruction-set processor / M. Stadler, T. Rower // Proceedings of the ITC International Test Conference. – 1999. – P. 414.

101. Hahanov V.I. Brain-like Computer Structures / V.I. Hahanov, S.V. Chumachenko, C.U.Ngene, Y. Tiesoura // Радиоэлектроника и информатика.– 2009.– № 4.– С. 30-40.

102. Хаханов В.И. Встроенное диагностирование цифровых систем / В.И. Хаханов, С.В. Чумаченко, Y. Tiesoura, С.С. Галаган // Радіоелектронні і комп'ютерні системи.– 2009.– №7(41).– С. 314-318.

103. Хаханов В.И. Параллельные мультипроцессорные процесс-модели векторно-логического анализа / В.И. Хаханов, С.В.Чумаченко, А.В. Хаханова, Y. Tiesoura // Информационно-управляющие системы на железнодорожном транспорте.– 2010.– № 4.– С. 51-57.

104. Хаханов В.И. Пример определения тестопригодности цифрового проекта / В.И. Хаханов, Е.И. Литвинова, И.А. Побеженко, Y. Tiesoura // Радіоелектронні і комп'ютерні системи.– 2010. №6(47).– С. 292-297.

105. Хаханов В.И. Тестирование и верификация HDL-моделей компонентов SOC. II / В.И. Хаханов, Е.И. Литвинова, И.А. Побеженко, Y. Tiesoura, С.U. Ngene // АСУ и приборы автоматики.– 2009.– Вып. 148.– С. 26-37.

106. Хаханов В.И. Инфраструктура логического анализа ассоциативных таблиц / В.И. Хаханов, О.А. Гузь, Ив Тиекура, К.У. Нгене (С.U. Ngene), В.И. Копанев // АСУ и приборы автоматики.– 2009.– Вып. 149.– С. 18-29.

107. Tiesoura Y. Актуальные проблемы анализа киберпространства / Yves Tiesoura, С.В. Чумаченко, В.И. Хаханов // АСУ и приборы автоматики.– 2011.– Вып. 154.– С. 59-75.

108. Тиекура И. Анализ киберпространства и диагностирование функциональных модулей / Ив Тиекура, В.И. Хаханов, С.В. Чумаченко, Е.И. Литвинова // АСУ и приборы автоматики.– 2011.– Вып. 155.– С. 4-12.

109. Хаханов В. Энергосберегающие технологии для программно-аппаратной имплементации систем вычислительного интеллекта / В. Хаханов, С.U. Ngene, Yves Tiesoura // Материалы межд. научн. конф.

«Интеллектуальные системы принятия решений и проблем вычислительного интеллекта» (ISDMCT09).– Евпатория, 2009.– Т.2.– С. 439-458.

110. Hahanov V.I. Embedded Testing for SoC Functionality // V.I. Hahanov, S. Pokrova, Y. Tiesoura, A.A. Gorobets // Proceedings of the Xth International Conference CADSM.– Lviv-Polyana.– 2009.– P. 29-33.

111. Hahanov V.I. Algebra-Logical Diagnosis and Repair Method for SoC Memory // V.I. Hahanov, E.I. Litvinova, O.A. Guz, Y. Tiesoura // Proceedings of IFAC Discrete-Event System Design.– Candia Beach, Spain.– 2009.– P. 15-20.

112. Hahanov V. Testing and Verification of HDL-models for SoC components / Vladimir Hahanov, Irina Hahanova, C.U. Ngene, Yves Tiesoura // Proc. of IEEE East-West Design and Test Symposium. – Russia, 2009.– P. 48-53.

113. Hahanov V. SoC Design Quality Increasing by Using Assertion Engine / Vladimir Hahanov, Irina Hahanova, Eugenia Litvinova, C.U. Ngene, Yves Tiesoura // Материалы VI Международной научно-технической конференции «Повышение качества, надежности и долговечности технических систем и технологических процессов».– Хургада, 2009. – С.124-126.

114. Hahanov V.I. IP-infrastructure learning for SoC service // V.I. Hahanov, O.A. Guz, W. Gharibi, Y. Tiesoura // V Международная конференция «Стратегия качества в промышленности и образовании».– Варна, Болгария.–2009.– С. 661-665.

115. Хаханов В.И. Инфраструктура мозгоподобных неарифметических вычислений / В.И. Хаханов, С.В.Чумаченко, Y.Tiesoura, O.A. Гузь // Материалы XV Международной научно-практической конференции.– 2010.– Алушта.– С. 69-75.

116. Хаханов В И. Алгебра для анализа векторно-логического пространства с помощью мультипроцессорного компьютера / В И. Хаханов, С.В Чумаченко, Y. Tiesoura, А. Мищенко // Матеріали міжнародної наукової конференції «Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту».– Євпаторія.– 2010.– С. 414-418.

117. Хаханов В.И. Метрика векторно-логического пространства и ее практическое применение / В.И. Хаханов, А.В. Хаханова, Yves Tiesoura, C.U. Ngene // Материалы конференции "Сучасні інформаційні та інноваційні технології на транспорті MINTT-2010".– Херсон, 2010.– С. 247-250.

118. Hahanov V.I. Logical method for detecting faults by fault detection table / V.I. Hahanov, I. Pobizhenko, Y. Tiesoura // Proceeding of IEEE East-West Design and Test Symposium.– St. Petersburg, Russia.– 2010.– С. 215-217.

119. Хаханов В. Логический ассоциативный анализ информации // В. Хаханов, С.В. Чумаченко, Y. Tiesoura, В.А. Василенко // Материалы Седьмой Международной конференции «Автоматизация проектирования дискретных систем».– Минск.– 2010.– С. 23-32.

120. Hahanov V.I. Algebra logical method of fault diagnosis / V.I. Hahanov, Y. Tiesoura, A. Pashchenko, V. Olhovoy // Матеріали міжнародної наукової конференції CADSM.– Lviv-Polyana.– 2011.– С. 186–187.

121. Хаханов В.И. Эволюционная модель киберпространства / В.И. Хаханов, С.В. Чумаченко, А.С. Мищенко, Y. Tiesoura // Матеріали VI міжнародної наукової конференції «Наука и социальные проблемы общества – информатизация и информационные технологии».– Харьков.– 2011.– С. 383-384.