

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кваліфікаційна наукова праця
на правах рукопису

ПАХОМОВ ЮРІЙ ВАСИЛЬОВИЧ

УДК 681.326:519.714

ДИСЕРТАЦІЯ

МОДЕЛІ ТА МЕТОДИ ТЕСТОПРИДАТНОГО ПРОЕКТУВАННЯ КРИТИЧНИХ СИСТЕМ ЛОГІЧНОГО УПРАВЛІННЯ НА ОСНОВІ КІНЦЕВИХ АВТОМАТІВ

05.13.05 - комп'ютерні системи та компоненти

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Ю.В. Пахомов

Науковий керівник: Мірошник Марина Анатоліївна,
доктор технічних наук, професор

Цей примірник дисертаційної роботи ідентичний за змістом
з іншими, поданими до спеціалізованої вченої ради Д 64.052.01

Учений секретар спеціалізованої
вченої ради Д 64.052.01

Є. І. Литвинова

Харків - 2018

АНОТАЦІЯ

Пахомов Юрій Васильович. Моделі та методи тестопридатного проектування критичних систем логічного управління на основі кінцевих автоматів. – Кваліфікаційна наукова робота на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук (доктора філософії) за спеціальністю 05.13.05 – комп'ютерні системи і компоненти. – Харківський національний університет радіоелектроніки, Міністерство освіти і науки України, Харків, 2018.

Мета дослідження: – розробка моделей і методів автоматизованого проектування та діагностування автоматних систем логічного управління на ПЛІС з використанням мов опису апаратури для скорочення часу технічного обслуговування критичних систем.

Задачі дослідження: 1) Визначити особливості діагностування критичних апаратних систем логічного управління. 2) Удосконалити методи неруйнівних діагностичних експериментів для моделей кінцевих автоматів. 3) Розробити методи побудови апаратних систем підтримки проведення діагностичних експериментів в автоматних системах керування. 4) Удосконалити методи оцінки тестопридатності графових моделей кінцевих автоматів. 5) Розробити методи підвищення тестопридатності моделей кінцевих автоматів за рахунок розширення вхідного алфавіту. 6) Реалізувати розроблені моделі в програмних модулях побудови тестопридатних автоматів в САПР ПЛІС.

Об'єкт дослідження: – процес забезпечення тестопридатності цифрових систем на основі кінцевих автоматів.

Предмет дослідження: – моделі та методи тестопридатного проектування і діагностування автоматних систем логічного управління на ПЛІС.

Науково-практична задача дослідження: – розробка моделей і методів автоматизованого проектування та діагностування тестопридатних цифрових систем логічного управління на основі кінцевих автоматів.

Сутність дослідження полягає в підвищенні надійності функціонування критичних систем логічного управління за рахунок скорочення часу та підвищення якості методів діагностування у рамках проведення регламентних робіт щодо технічного обслуговування апаратної частини локальних систем управління технологічно відокремлених об'єктів електроенергетики та газопостачання, побудованих на основі кінцевих цифрових автоматів. Скорочення часу та підвищення якості методів діагностування досягається за рахунок розробки методів проведення тестопридатного проектування та діагностування автоматних систем логічного управління, побудованих на технологічній платформі програмованих логічних інтегральних схем. Тестопридатне проектування здійснюється з використанням інструментальних засобів систем автоматизованого проектування на основі мов опису апаратури.

Методи дослідження – апарати булевої алгебри, теорії автоматів і графів (подання моделей цифрових автоматів), технічної діагностики (підготовка та проведення діагностичних експериментів), засоби автоматизованого тестопридатного проектування цифрових автоматів (побудова, моделювання та синтез автоматних HDL-моделей).

Наукова новизна результатів дослідження:

1) *вперше* запропоновано розширення вхідного алфавіту кінцевого автомата, яке характеризується введенням додаткового стовпця в таблицю переходів-виходів автомата, *що суттєво скоротило довжину та час* діагностичного експерименту за рахунок можливості встановлювати автомат в довільний стан;

2) *отримали подальший розвиток* методи проведення діагностичного експерименту за рахунок використання синхронізуючих послідовностей, *що дозволяє проводити* неруйнівний діагностичний експеримент навіть для автоматів з несправностями функцій переходів;

3) *отримали подальший розвиток* методи побудови апаратних пристроїв діагностування за рахунок реалізації стратегії обходу всіх станів

або дуг графа переходів керуючого автомата, *що дозволяє проводити діагностичні експерименти*, не порушуючи основний режим роботи критичної системи логічного управління протягом тривалого часу;

4) *удосконалено методи* розрахунку тестопридатності кінцевих керуючих автоматів, *які, на відміну від існуючих методів*, дозволили мінімізувати додаткові апаратні витрати при побудові легкотестованих автоматів.

Ринкова привабливість дослідження. Впровадження моделей та методів тестопридатного проектування критичних систем логічного управління на основі керуючих автоматів приведе до зменшення часу на проектування і технічне обслуговування автоматних систем логічного управління та кількості обслуговуючого персоналу.

Практична значення отриманих результатів досліджень полягає у розробці методики, що дозволяє будувати тестопридатні моделі кінцевих автоматів з використанням мов опису апаратури САПР РЕП та шаблонів автоматного програмування. Розроблені процедури розрахунку тестопридатності кінцевих керуючих автоматів дали можливість оптимізувати створення додаткових переходів у графових моделях автоматів, що дозволило зменшити апаратні витрати при побудові легкотестованих керуючих автоматів. Розроблено програмний модуль із візуальним інтерфейсом для автоматизованого проектування тестопридатних HDL-моделей кінцевих керуючих автоматів. Програмний модуль можна реалізувати у вигляді хмарного сервісу.

Практична реалізація моделей і методів тестопридатного проектування критичних систем логічного управління на основі кінцевих автоматів виконана в рамках створення й верифікації програмних компонентів із візуальним інтерфейсом для автоматизованого проектування тестопридатних HDL-моделей кінцевих керуючих автоматів.

Результати дисертації у складі моделей, методів і фрагментів програмних застосунків упроваджено в навчальному процесі Харківського

національного університету радіоелектроніки (акт про впровадження від 21.12.2018).

Розроблено HDL-моделі, які моделюються та синтезуються інструментальними засобами систем автоматизованого проектування цифрових пристроїв на технологічній платформі ПЛІС. Розроблено програмний модуль із візуальним інтерфейсом для вводу графа переходів керуючого автомата для автоматизованої побудови тестопридатних HDL-моделей автоматів у формі автоматного шаблону в синтезованій підмножині мови опису апаратури VHDL, який був застосований у підрозділах ПрАТ «Енергооблік». Це підприємство є розробником мікропроцесорних вимірювальних приладів та систем тестопридатного проектування (довідка про впровадження від 28.11.2018).

Розроблено методики апаратного діагностування локальних пристроїв керування та регулювання технологічно відокремлених об'єктів електроенергетики та газопостачання. Алгоритми керування описуються граф-схемою алгоритму, яка представляється математичною моделлю графа переходів кінцевого автомата. Для реалізації алгоритму діагностування використовується спосіб обходу всіх дуг графа та моделі керуючого автомата й пристрою діагностування, які описуються мовою опису апаратури VHDL з наступним синтезом та використанням пристроїв програмованої логіки. Ці пристрої були застосовані в підрозділах АТ «Укртрансгаз» для створення систем контролю на автоматичних газорозподільчих станціях, обладнаних сучасними автоматичними приладами, розробленими на базі цифрової техніки (довідка про впровадження від 25.10.2018).

Результати дисертаційної роботи у вигляді автоматної моделі апаратної системи діагностування цифрових пристроїв управління в електроенергетиці й газопостачанні також було застосовано в діяльності підрозділів ПАТ «Харківміськгаз» для створення надійних систем контролю в міських автоматизованих газорозподільчих пунктах, які оснащені сучасним автоматичним обладнанням на базі цифрової техніки. Це дозволило суттєво

скоротити час на виявлення та усунення можливих дефектів, які виникають на об'єктах газотранспортної системи, та підвищити ступінь надійності транспортування газу.

Впровадження результатів дослідження у виробничу діяльність ПАТ «Харківміськгаз» (довідка про впровадження від 15.11.2018) забезпечило ефективність вимірів технічних параметрів роботи технологічного обладнання ГРП та вузлів обліку газу, що дозволило аналізувати залежність зміни одних параметрів від інших і, як наслідок, підвищення надійності газопостачання, а також спостереження тенденції щодо скорочення комерційних втрат газу.

Публікації. Результати дисертаційної роботи відображено у 26 друкованих працях, серед яких 1 монографія, 13 статей у наукових журналах із Переліку наукових фахових видань України, 3 статті в міжнародних наукових журналах за кордоном, 5 статей входять до міжнародної наукометричної бази Scopus; 1 стаття – Web of Science; 9 матеріалів міжнародних наукових конференцій (з них 3 за кордоном і 2 входять до наукометричної бази Scopus). Здобувачеві належать 6 публікацій, що входять до наукометричної бази Scopus, індекс Хірша $h=1$.

<https://www.scopus.com/authid/detail.uri?authorId=57190816915#top>

СПИСОК ОПУБЛІКОВАНИХ РОБІТ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

в яких відображені основні наукові результати дисертації:

1. *Пахомов Ю. В.* Технологія ремонту газового обладнання і трубопроводних систем, монографія / І. І. Капцов, В. Г. Котух, Ю. В. Пахомов. – Харків : ХНУМГ ім. О. М. Бекетова, 2016. – 232 с.

2. *Пахомов Ю. В.* Проведення діагностичних експериментів у керуючих автоматах з використанням синхронізуючих послідовностей / [М. А. Мірошник, О. С. Шкіль, Ю. В. Пахомов та ін.] // *Радіоелектроніка та інформатика: наук.-техн. журнал.* – Харків: ХНУРЕ, 2018. – № 3 – С. 82-89 (Реферується або індексується міжнародними наукометричними базами Index

Copernicus (<http://journals.indexcopernicus.com/-p24787015.3.html>), Google Scholar, CiteFactor, NBUV, SIS, OAJI.net, Cyberleninka, OECSP, Scholar Steer, TIU Hannover, I2OR).

3. *Pakhomov Y.* Design automation of easy-tested digital finite state machines / [М. А. Miroshnik, Y. V. Pakhomov, A. S. Shkil та ін.] // Radio Electronics, Computer Science, Control, the scientific journal, Zaporizhzhia National Technical University. – 2018. – № 2. – P. 117-124 (Реферується або індексується міжнародними наукометричними базами Thomson Reuters Web of Science (WoS), CiteFactor, COPAC, CrossRef, eLibrary.ru / РИНЦ, GENERAL IMPACT FACTOR, Google Scholar, Impactfactor.pl, Index Copernicus, Scholar Steer, SIS, SSM).

4. *Пахомов Ю. В.* Аналіз апаратурних витрат при тестопридатному проектуванні керуючих цифрових автоматів / [М. А. Мірошник, Ю. В. Пахомов, О. С. Шкіль та ін.] // Вісник СХУ ім. В. Даля. – 2018. – № 6. – С. 101-109 (Реферується або індексується міжнародними наукометричними базами eLibrary.ru, NBUV, WorldWideScience.org, ScienceDirect, Google Scholar, WorldCat, BASE, DOAJ, URAN).

5. *Pakhomov Y. V.* Model of automated hardware diagnostics of remote energy systems management points / М. А. Miroshnyk, Y. V. Pakhomov // Світлотехніка та електроенергетика: міжнар. наук.-техн. журнал. – Харків: ХНУМГ ім. О. М. Бекетова. – 2017. – № 3. – С. 3-9 (Реферується або індексується міжнародними наукометричними базами Index Copernicus, Google Scholar, WorldCat).

6. *Пахомов Ю. В.* Методы проектирования самопроверяемых цифровых автоматов / [М. А. Мировшник, Э. Н. Кулак, Ю. В. Пахомов та ін.] // Радиотехника: всеукр. межвед. науч.-техн. зб. – Харків: ХНУРЕ, 2016. – № 187. – С. 124-131 (Реферується або індексується міжнародними наукометричними базами Google Scholar, НБУВ, Elibrary.ru).

7. *Пахомов Ю. В.* Методы синтеза легкотестируемых цифровых автоматов / А. С. Гребенюк, М. А. Мировшник, Ю. В. Пахомов,

И. В. Филиппенко // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. – Харків: УкрДУЗТ. – 2016. – № 5. – С. 28-39 (Реферується або індексується міжнародними базами Google Scholar, РИНЦ, Elibrary.ru).

які засвідчують апробацію матеріалів дисертації:

8. *Pakhomov Y.* Design of Logical Control Units Based on Finite State Machines' Patterns / [M. Miroshnyk, Y. Pakhomov, A. Shkil et al.] // IEEE East-West Design & Test Symposium (EWDTS'2018, Kazan, Russia, 14-17 Sept. 2018) – 6 p (Indexed by Scopus).

9. *Pakhomov Y. V.* Microware multiport multimeter sensor mutual reflection and its influence on signal and tract parameter measurement accuracy / [M. A. Miroshnyk, O. B. Zaichenko, Y. V. Pakhomov et al.] // 28-th International Scientific Symposium "Metrology and Metrology Assurance" (Sozopol, Bulgaria, 10-14 Sept. 2018). – P. 60-64 (Реферується або індексується міжнародними наукометричними базами Elibrari.ru, Erih plus, Scince Index, DOAJ, РИНЦ).

10. *Пахомов Ю. В.* Проектирование логических блоков управления с помощью шаблонов описания конечных автоматов / М. А. Мирошник, Ю. Н. Салфетникова, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: 31-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 24-26 жовт. 2018р.). – № 4. – С. 21-22 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РИНЦ, Elibrary.ru).

11. *Pakhomov Y.* Design automation of testable finite state machines / [M. Miroshnyk, Y. Pakhomov, S. Shkil et al.] // IEEE East-West Design & Test Symposium (EWDTS'2017, Novi Sad, Serbia, Sept. 27-Oct. 2, 2017). – P 203-208 (Indexed by Scopus, IEEE Xplore).

12. *Пахомов Ю. В.* Исследование методов синтеза легкотестируемых цифровых устройств и систем / М. А. Мирошник, В. А. Крылова, Ю. В. Пахомов, А. Н. Мирошник // Проблеми інформатики та моделювання (ПІМ-2017): 17-та міжнар. наук.-техніч. конф. (Харків, НТУ «ХП»,

11-15 вер. 2017р.) С. 58-59 (Реферується або індексується міжнародними наукометричними базами Index Copernicus, Google Scholar, OAJ, Cyberleninka, WorldCat, BASE, UIF, DOAJ, NBUV MIAR, DOI, CrossRef).

13. *Пахомов Ю. В.* Методы обнаружения ошибок проектирования в конечных автоматах с использованием синхронизирующих последовательностей / М. А. Мирошник, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: 30-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 26-27 жовт. 2017р.). – № 4. – С. 15-16 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, Elibrary.ru).

14. *Пахомов Ю. В.* Методи оцінки ефективності ремонтно-відновлювальних робіт для виробів газового обладнання та трубопровідних систем / М. А. Мірошник, В. Г. Котух, Ю. В. Пахомов. // Інформаційні технології та комп'ютерна інженерія: 5-та міжнар. наук.-практ. конф. (Івано-Франківськ, Вінниця, 27-29 трав. 2015р.). – С. 15-16 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, DOAJ).

15. *Пахомов Ю. В.* Структурный анализ измерительных систем датчиков для газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Інформаційні технології та комп'ютерна інженерія: 4-та міжнар. наук.-практ. конф. (Вінниця, 28-30 трав. 2014р.). – С. 32-34 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, DOAJ).

16. *Пахомов Ю. В.* Системный подход к определению состава вспомогательного материала для изделий газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: 27-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 24-26 вер. 2014р.). – № 4. – С. 30-31 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, Elibrary.ru).

які додатково відображають наукові результати дисертації:

17. *Pakhomov Y.* Information model of registration and analysis of technological factors arising during final processing of products of transport pipeline systems / V. Kotukh, N. Kaptsova, Y. Pakhomov, V. Kosenko // International Journal of Engineering and Technology (UAE, ISSN: 2227-524X). – 2018. – № 2.23. – С. 73-76 (Indexed by Scopus, IEEE Xplore).

18. *Пахомов Ю. В.* Методы автоматизации проектирования легкотестируемых компьютерных систем и устройств на основе цифровых автоматов / М. А. Мирошник, Л. А. Клименко, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. – 2018. – № 4. – С. 3-10 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, Elibrary.ru).

19. *Pakhomov Y.* Methods for Designing Self-Checking Digital Machines / [M. Miroschnyk, E. Kulak, Y. Pakhomov та ін.] // Telecommunications and Radio Engineering, USA. – 2017. – № 15. – С. 1367-1377 (Indexed by Scopus, IEEE Xplore).

20. *Pakhomov Y.* Konstruktion of Distributed Information Management Systems for Accounting and Control of Energy Consumption at the Example of Gas / V. Kotukh, Y. Pakhomov // Telecommunications and Radio Engineering, USA. – 2016. – № 7. – С. 631-641 (Indexed by Scopus, IEEE Xplore).

21. *Пахомов Ю. В.* Разработка алгоритма прогнозирования функционально-технического состояния изделий газового оборудования и трубопроводных систем / Ю. В. Пахомов // Вестник БГТУ им. В. Г. Шухова: науч.-теор. журнал. – 2015. – № 3. – С. 90-97 (Реферується або індексується міжнародними наукометричними базами, Elibrary.ru, РІНЦ, ІАС, Science index).

22. *Пахомов Ю. В.* Основные принципы создания единой распределенной системы автоматического контроля и учета энергоресурсов на примере газовой отрасли / В. Г. Котух, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. –

Харків: УкрДУЗТ, 2015. – № 3. – С. 48-55 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, http://elibrary.ru/title_about.asp?id=33934).

23. *Пахомов Ю. В.* Построение распределенных информационно-управляющих систем учета и контроля энергоресурсов на примере газовой отрасли / В. Г. Котух, Ю. В. Пахомов // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харків: ХНУРЕ, 2015. – № 182.– С. 65-72 (Реферується або індексується міжнародними наукометричними базами Google Scholar, НБУВ, eLIBRARY.RU).

24. *Пахомов Ю. В.* Исследование распределения тепловых потоков при импульсной лазерной сварке корпусов датчиков для газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Вестник БГТУ им. В. Г. Шухова. – 2014. – № 5. – С. 96-101 (Реферується або індексується міжнародними базами НЕБ Elibrary.ru, РІНЦ, IAC, Science index).

25. *Пахомов Ю. В.* Технологическая концепция создания АСУ ТП для объектов энергоснабжения на базе цифровых распределенных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харків: ХНУРЕ, 2014. – № 179. – С. 131-137 (Реферується або індексується міжнародними базами Google Scholar, НБУВ, Elibrary.ru).

26. *Пахомов Ю. В.* Методы расчета упругих элементов мембранно-балочного типа в датчиках для газового оборудования и трубопроводных систем / Ю. В. Пахомов // Комунальне господарство міст: наук.-техн. зб. – Харків: ХНУМГ ім. О. М. Бекетова, 2014. – № 116. – С. 64-66 (Реферується або індексується міжнародними наукометричними базами Google Scholar, WorldCat, DOAJ, НБУВ).

Ключові слова: критична система, керуючий автомат, проектування, діагностування, тестопридатність, апаратурні витрати, мови опису апаратури, САПР ПЛІС.

АННОТАЦИЯ

Пахомов Юрий Васильевич. Модели и методы тестопригодного проектирования критических систем логического управления на основе конечных автоматов. – Квалификационная научная работа на правах рукописи.

Диссертация на соискание ученой степени кандидата технических наук по специальности 05.13.05 – компьютерные системы и компоненты. – Харьковский национальный университет радиоэлектроники, Министерство образования и науки Украины, Харьков, 2019.

Цель исследования – разработка моделей и методов автоматизированного проектирования и диагностирования автоматных систем логического управления на ПЛИС с использованием языков описания аппаратуры для сокращения времени технического обслуживания карточных систем.

Задачи исследования: 1) Определить особенности диагностирования критических аппаратных систем логического управления. 2) Усовершенствовать методы неразрушающих диагностических экспериментов для моделей конечных автоматов. 3) Разработать методы построения аппаратных систем поддержки проведения диагностических экспериментов в автоматных системах управления. 4) Усовершенствовать методы оценки тестопригодности графовых моделей конечных автоматов. 5) Разработать методы повышения тестопригодности моделей конечных автоматов за счет расширения входного алфавита. 6) Реализовать разработанные модели в программных модулях построения тестопригодных автоматов в САПР ПЛИС.

Объект исследования: – процесс обеспечения тестопригодности цифровых систем на основе конечных автоматов.

Предмет исследования: – модели и методы тестопригодного проектирования и диагностирования автоматных систем логического управления на ПЛИС.

Научно-практическая задача исследования – разработка моделей и методов автоматизированного проектирования и диагностирования тестопригодных цифровых систем логического управления на основе конечных автоматов. Сущность исследования заключается в повышении надежности функционирования критических систем логического управления за счет сокращения времени и повышения качества методов диагностики в рамках проведения регламентных работ по обслуживанию аппаратной части локальных систем управления технологически обособленных объектов электроэнергетики и газоснабжения, построенных на основе конечных цифровых автоматов. Сокращение времени и повышение качества методов диагностирования достигается за счет разработки методов проведения тестопригодного проектирования и диагностирования автоматных систем логического управления, построенных на технологической платформе программируемых логических интегральных схем. Тестопригодное проектирование осуществляется с использованием инструментальных средств систем автоматизированного проектирования на основе языков описания аппаратуры.

Методы исследования – аппараты булевой алгебры, теории автоматов и графов (представление моделей цифровых автоматов), технической диагностики (подготовка и проведение диагностических экспериментов), средства автоматизированного тестопригодного проектирования цифровых автоматов (построение, моделирование и синтез автоматных HDL-моделей).

Научная новизна исследования:

1) впервые предложено расширение входного алфавита конечного автомата, характеризующееся введением дополнительного столбца в таблицу переходов-выходов автомата, что существенно сократило длину и время диагностического эксперимента за счет возможности установки автомата в произвольное состояние;

2) получили дальнейшее развитие методы проведения диагностического эксперимента за счет использования синхронизирующих

последовательностей, что позволило проводить неразрушающий диагностический эксперимент даже для автоматов с неисправностями функций переходов;

3) получили дальнейшее развитие методы построения аппаратных устройств диагностирования за счет реализации стратегии обхода всех состояний или дуг графа переходов управляющего автомата, что позволило проводить диагностические эксперименты, не нарушая основной режим работы критической системы логического управления в течение длительного времени;

4) усовершенствованы методы расчета тестопригодности конечных управляющих автоматов, которые, в отличие от существующих методов, позволили минимизировать дополнительные аппаратные затраты при построении легкотестируемых цифровых автоматов.

Рыночная привлекательность исследования. Внедрение моделей и методов тестопригодного проектирования критических систем логического управления на основе управляющих автоматов приведет к уменьшению времени на проектирование и техническое обслуживание автоматных систем логического управления и количества обслуживающего персонала.

Практическая значимость полученных результатов исследований заключается в разработке методики, позволяющей строить тестопригодные модели конечных автоматов с использованием языков описания аппаратуры САПР РЭП и шаблонов автоматного программирования. Разработанные процедуры расчета тестопригодности конечных управляющих автоматов позволили оптимизировать создание дополнительных переходов в графовых моделях автоматов, что позволило уменьшить аппаратные затраты при построении легкотестируемых управляющих автоматов. Разработан программный модуль с визуальным интерфейсом для автоматизированного проектирования тестопригодных HDL-моделей конечных управляющих автоматов. Программный модуль можно реализовать в виде облачного сервиса.

Практическая реализация моделей и методов тестопригодного проектирования критических систем логического управления на основе конечных автоматов выполнена в рамках создания и верификации программных компонентов с визуальным интерфейсом для автоматизированного проектирования тестопригодных HDL-моделей конечных управляющих автоматов.

Результаты диссертации в составе моделей, методов и фрагментов программных приложений внедрены в учебном процессе Харьковского национального университета радиоэлектроники (акт о внедрении от 21.12.2018).

Разработаны HDL-модели, которые моделируются и синтезируются инструментальными средствами автоматизированного проектирования цифровых устройств на технологической платформе ПЛИС. Разработан программный модуль с визуальным интерфейсом для ввода графа переходов управляющего автомата для автоматизированного построения тестопригодных HDL-моделей автоматов в форме автоматного шаблона в синтезируемом подмножестве языка описания аппаратуры VHDL, который был применен в подразделениях ЧАО «Энергоучет». Это предприятие является разработчиком микропроцессорных измерительных приборов и систем тестопригодного проектирования (справка о внедрении от 28.11.2018).

Разработаны методики аппаратного диагностирования локальных устройств управления и регулирования технологически обособленных объектов электроэнергетики и газоснабжения. Алгоритмы управления описываются граф-схемой алгоритма, которая представляется математической моделью графа переходов конечного автомата. Для реализации алгоритма диагностирования используется способ обхода всех дуг графа, модели управляющего автомата и устройства диагностирования, которые описываются на языке описания аппаратуры VHDL с последующим синтезом и использованием устройств программируемой логики. Эти

устройства были применены в подразделениях АО «Укртрансгаз» для создания систем контроля на автоматических газораспределительных станциях, оборудованных современными автоматическими приборами, разработанными на базе цифровой техники (справка о внедрении от 25.10.2018).

Результаты диссертационной работы в виде автоматной модели аппаратной системы диагностирования цифровых устройств управления в электроэнергетике и газоснабжении были применены в деятельности ПАО «Харьковгоргаз» для создания надежных систем контроля в городских автоматизированных газорегуляторных пунктах (ГРП), которые оснащены современным автоматическим оборудованием на базе цифровой техники. Это позволило существенно сократить время на выявление и устранение возможных дефектов, возникающих на объектах газотранспортной системы, и повысить степень надежности транспортировки газа.

Внедрение результатов исследования в производственную деятельность ПАО «Харьковгоргаз» (справка о внедрении от 15.11.2018) обеспечило эффективность измерений технических параметров работы технологического оборудования ГРП и узлов учета газа, что позволило анализировать зависимость изменения одних параметров от других и, как следствие, повысить надежность работы системы газоснабжения с тенденцией сокращения коммерческих потерь газа.

Публикации. Результаты диссертационной работы отражены в 26 печатных работах, среди которых 1 монография, 13 статей в научных журналах из Перечня научных профессиональных изданий Украины, 3 статьи в международных научных журналах за рубежом, 5 статей входят в международную наукометрическую базу Scopus; 1 статья – Web of Science; 9 материалов международных научных конференций (из них 3 за рубежом и 2 входят в наукометрическую базу Scopus). Соискателю принадлежат 6 публикаций, которые входят в наукометрическую базу Scopus, индекс Хирша $h = 1$. <https://www.scopus.com/authid/detail.uri?authorId=57190816915#top>

СПИСОК ОПУБЛИКОВАНИХ РАБОТ ПО ТЕМЕ ДИССЕРТАЦИИ

Список публикаций соискателя, в которых отражены основные научные результаты диссертации:

1. Пахомов Ю. В. Технологія ремонту газового обладнання і трубопровідних систем, монографія / І. І. Капцов, В. Г. Котух, Ю. В. Пахомов. – Харків : ХНУМГ ім. О. М. Бекетова, 2016. – 232 с.

2. Пахомов Ю. В. Проведення діагностичних експериментів у керуючих автоматах з використанням синхронізуючих послідовностей / [М. А. Мірошник, О. С. Шкіль, Ю. В. Пахомов та ін.] // Радіоелектроніка та інформатика: наук.-техн. журнал. – Харків: ХНУРЕ, 2018. – № 3 – С. 82–89 (Реферується або індексується міжнародними наукометричними базами Index Copernicus (<http://journals.indexcopernicus.com/-p24787015.3.html>), Google Scholar, CiteFactor, NBUV, SIS, OAJI.net, Cyberleninka, OECSP, Scholar Steer, TIU Hannover, I2OR).

3. Pakhomov Y. Design automation of easy-tested digital finite state machines / [M. A. Miroshnik, Y. V. Pakhomov, A. S. Shkil et al.] // Radio Electronics, Computer Science, Control, the scientific journal, Zaporizhzhia National Technical University. – 2018. – № 2. – С. 117–124 (Реферується або індексується міжнародними наукометричними базами Thomson Reuters Web of Science (WoS), CiteFactor, COPAC, CrossRef, eLibrary.ru / РИНЦ, GENERAL IMPACT FACTOR, Google Scholar, Impactfactor.pl, Index Copernicus, Scholar Steer, SIS, SSM).

4. Пахомов Ю. В. Аналіз апаратурних витрат при тестопридатному проектуванні керуючих цифрових автоматів / [М. А. Мірошник, Ю. В. Пахомов, О. С. Шкіль та ін.] // Вісник СХУ ім. В. Даля. – 2018. – № 6. – С. 101–109 (Реферується або індексується міжнародними наукометричними

базами eLibrary.ru, NBUV, WorldWideScience.org, ScienceDirect, Google Scholar, WorldCat, BASE, DOAJ, URAN).

5. Pakhomov Y. V. Model of automated hardware diagnostics of remote energy systems management points / M. A. Miroshnyk, Y. V. Pakhomov // Світлотехніка та електроенергетика: міжнародний науково-технічний журнал. – Харків: ХНУМГ ім. О. М. Бекетова, 2017. – № 3. – С. 3–9 (Реферується або індексується міжнародними наукометричними базами Index Copernicus, Google Scholar, WorldCat).

6. Пахомов Ю. В. Методы проектирования самопроверяемых цифровых автоматов / [М. А. Мирошник, Э. Н. Кулак, Ю. В. Пахомов и др.] // Радиотехника: всеукр. межвед. науч.-техн. зб. – Харків: ХНУРЕ, 2016. – № 187. – С. 124–131 (Реферується або індексується міжнародними наукометричними базами Google Scholar, НБУВ, Elibrary.ru).

7. Пахомов Ю. В. Методы синтеза легкотестируемых цифровых автоматов / А. С. Гребенюк, М. А. Мирошник, Ю. В. Пахомов, И. В. Филиппенко // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. – Харків: УкрДУЗТ, 2016. – № 5. – С. 28–39 (Реферується або індексується міжнародними базами Google Scholar, РІНЦ, Elibrary.ru).

Публикации, в которых отражены результаты апробации материалов диссертации:

8. Pakhomov Y. Design of Logical Control Units Based on Finite State Machines' Patterns / [M. Miroshnyk, Y. Pakhomov, A. Shkil et al.] // IEEE East-West Design & Test Symposium (EWDTS'2018, Kazan, Russia, 14-17 Sept. 2018) – 6 p (Indexed by Scopus).

9. Pakhomov Y. V. Microwave multiport multimeter sensor mutual reflection and its influence on signal and tract parameter measurement accuracy / [M. A. Miroshnyk, O. B. Zaichenko, Y. V. Pakhomov et al.] // 28-th International

Scientific Symposium "Metrology and Metrology Assurance" (Sozopol, Bulgaria, 10–14 Sept. 2018). – P. 60–64 (Реферується або індексується міжнародними наукометричними базами Elibrari.ru, Erih plus, Scince Index, DOAJ, РИНЦ).

10. Пахомов Ю. В. Проектирование логических блоков управления с помощью шаблонов описания конечных автоматов / М. А. Мирошник, Ю. Н. Салфетникова, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: 31-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 24–26 жовт. 2018р.). – № 4. – С. 21–22 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РИНЦ, Elibrary.ru).

11. Pakhomov Y. Design automation of testable finite state machines / [M. Miroshnyk, Y. Pakhomov, S. Shkil et al.] // IEEE East-West Design & Test Symposium (EWDTS'2017, Novi Sad, Serbia, Sept. 27–Oct. 2, 2017). – P 203-208 (Indexed by Scopus, IEEE Xplore).

12. Пахомов Ю. В. Исследование методов синтеза легкотестируемых цифровых устройств и систем / М. А. Мирошник, В. А. Крылова, Ю. В. Пахомов, А. Н. Мирошник // Проблеми інформатики та моделювання (ПІМ-2017): 17-та міжнар. наук.-техніч. конф. (Харків, НТУ «ХП», 11–15 вер. 2017р.) С. 58–59 (Реферується або індексується міжнародними наукометричними базами Index Copernicus, Google Scholar, ОАІ, Cyberleninka, WorldCat, BASE, UIF, DOAJ, NBUV MIAR, DOI, CrossRef).

13. Пахомов Ю. В. Методы обнаружения ошибок проектирования в конечных автоматах с использованием синхронизирующих последовательностей / М. А. Мирошник, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: 30-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 26–27 жовт. 2017р.). – № 4. – С. 15–16 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РИНЦ, Elibrary.ru).

14. Пахомов Ю. В. Методи оцінки ефективності ремонтно-відновлювальних робіт для виробів газового обладнання та трубопровідних систем / М. А. Мірошник, В. Г. Котух, Ю. В. Пахомов. // Інформаційні

технології та комп'ютерна інженерія: 5-та міжнар. наук.-практ. конф. (Івано-Франківськ, Вінниця, 27–29 трав. 2015р.). – С. 15–16 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, DOAJ).

15. Пахомов Ю. В. Структурный анализ измерительных систем датчиков для газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Інформаційні технології та комп'ютерна інженерія: 4-та міжнар. наук.-практ. конф. (Вінниця, 28–30 трав. 2014р.). – С. 32–34 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, DOAJ).

16. Пахомов Ю. В. Системный подход к определению состава вспомогательного материала для изделий газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: 27-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 24–26 вер. 2014р.). – № 4. – С. 30–31 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, Elibrary.ru).

Публикации, которые дополнительно отражают научные результаты диссертации:

17. Pakhomov Y. Information model of registration and analysis of technological factors arising during final processing of products of transport pipeline systems / V. Kotukh, N. Kaptsova, Y. Pakhomov, V. Kosenko // International Journal of Engineering and Technology (UAE, ISSN: 2227-524X). – 2018. – № 2.23. – С. 73–76 (Indexed by Scopus, IEEE Xplore).

18. Пахомов Ю. В. Методы автоматизации проектирования легкотестируемых компьютерных систем и устройств на основе цифровых автоматов / М. А. Мирошник, Л. А. Клименко, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: науково-техн.

журнал. – 2018. – №4 (131). – С. 3–10 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, Elibrary.ru).

19. Pakhomov Y. Methods for Designing Self-Checking Digital Machines / [M. Miroshnyk, E. Kulak, Y. Pakhomov et al.] // Telecommunications and Radio Engineering, USA. – 2017. – № 15. – С. 1367– 1377 (Indexed by Scopus, IEEE Xplore).

20. Pakhomov Y. Konstruktion of Distributed Information Management Systems for Accounting and Control of Energy Consumption at the Example of Gas / V. Kotukh, Y. Pakhomov // Telecommunications and Radio Engineering, USA. – 2016. – № 7. – С. 631– 641 (Indexed by Scopus, IEEE Xplore).

21. Пахомов Ю. В. Разработка алгоритма прогнозирования функционально-технического состояния изделий газового оборудования и трубопроводных систем / Ю. В. Пахомов // Вестник БГТУ им. В. Г. Шухова: науч.-теор. журнал. – 2015. – № 3. – С. 90–97 (Реферується або індексується міжнародними наукометричними базами, Elibrary.ru, РІНЦ, ІАС, Science index).

22. Пахомов Ю. В. Основные принципы создания единой распределенной системы автоматического контроля и учета энергоресурсов на примере газовой отрасли / В. Г. Котух, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. – Харків: УкрДУЗТ, 2015. – № 3. – С. 48–55 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, http://elibrary.ru/title_about.asp?id=33934).

23. Пахомов Ю. В. Построение распределенных информационно-управляющих систем учета и контроля энергоресурсов на примере газовой отрасли / В. Г. Котух, Ю. В. Пахомов // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харків: ХНУРЕ, 2015. – № 182.– С. 65–72 (Реферується або індексується міжнародними наукометричними базами Google Scholar, НБУВ, eLIBRARY.RU).

24. Пахомов Ю. В. Исследование распределения тепловых потоков при импульсной лазерной сварке корпусов датчиков для газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Вестник БГТУ им. В. Г. Шухова. – 2014. – № 5. – С. 96–101 (Реферується або індексується міжнародними базами НЕБ Elibrary.ru, РІНЦ, ІАС, Science index).

25. Пахомов Ю. В. Технологическая концепция создания АСУ ТП для объектов энергоснабжения на базе цифровых распределенных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харків: ХНУРЕ, 2014. – № 179. – С. 131–137 (Реферується або індексується міжнародними базами Google Scholar, НБУВ, Elibrary.ru).

26. Пахомов Ю. В. Методы расчета упругих элементов мембранно-балочного типа в датчиках для газового оборудования и трубопроводных систем / Ю. В. Пахомов // Комунальне господарство міст: наук.-техн. зб. – Харків: ХНУМГ ім. О. М. Бекетова, 2014. – № 116. – С. 64–66 (Реферується або індексується міжнародними наукометричними базами Google Scholar, WorldCat, DOAJ, НБУВ).

Ключевые слова: критическая система, управляющий автомат, проектирование, диагностирование, тестопригодность, аппаратурные затраты, языки описания аппаратуры, САПР ПЛИС.

ABSTRACT

Pahomov Yurii Vasylovyh. Models and methods for the testable design of critical systems of logic control based on finite-state machines. – Qualification work as a manuscript.

PhD thesis (candidate degree of technical sciences) in speciality 05.13.05 – Computer Systems and Components. – Kharkiv National University of Radio Electronics, Ministry of Education and Science of Ukraine, Kharkiv, 2018.

The PhD thesis is devoted to the development of models and methods of computer-aided design and diagnosis of automaton logic control systems on FPGA using hardware description languages.

Main results of the research:

1. The extension of the input alphabet of the finite state machine is proposed by introducing an additional column into the table of transitions-outputs of the automaton and an additional input to the circuit implementation of the control automaton.

2. The procedures for performing a diagnostic experiment are improved through the use of synchronization sequences.

3. The methods for developing hardware diagnostics devices have been improved; they implement a strategy for bypassing all the states or arcs of the transition graph of a control automaton by simulating the algorithm of an operational automaton in a logical control system.

4. The methods for calculating the testability of finite state machine, controllability, observability and the reachability of nodes (states) of a transition graph of control automata have been improved.

5. A new approach to computer-aided design of testable finite state machines has been proposed, based on the use of hardware description languages for describing models (HDL models), in which additional transitions are implemented by adding conditional statements to the VHDL description of transition functions and automaton pattern. The developed HDL descriptions are simulated and

synthesized by computer-aided design tools based on the FPGA technology platform.

6. The developed procedures for calculating the testability of the control automaton made it possible to optimize of entering additional transitions in automaton models through calculating the Quine estimate of the synthesized automaton models.

7. A software module has been developed; it has GUI for creating the transition graph of the control automaton and automating the construction of testable HDL models of automata in the form of automaton pattern in a synthesized subset of the VHDL language.

The research results obtained in the course of the research are reliable, which is confirmed by the experiments carried out to verify the proposed models and methods for monitoring and controlling gas supply systems.

The results of the thesis are reflected in 26 publications: 1 monograph, 13 articles that are included in the “Lists of scientific professional editions of Ukraine”, 3 articles in international scientific journals abroad (5 of them in the international scientific database Scopus, 1 in the international scientific database Thomson Reuters Web of Science); 9 publications at international scientific conferences (3 of them are abroad, 2 are part of the scientific databases Scopus and IEEE Xplore). The applicant has 6 publications included in the Scopus Science Center, and has the Hirsch index $h = 1$.

Key words: critical system, control finite state machine, design, diagnosis, testability, hardware costs, hardware description languages, CAD FPGA.

LIST OF PUBLICATIONS

The list of publications, which reflect the main scientific results of the thesis:

1. *Pakhomov Y. V.* Tekhnologiya remontu gazovogo obladnannya i truboprovodnykh system / I. I. Kaptsov, V. G. Kotukh, Y. V. Pakhomov. – Kharkiv, KhNUMG im. O.M. Beketova, 2016. – 232 s.

2. *Pakhomov Y. V.* Provedennya diahnostychnykh eksperymentiv u keruyuchykh avtomatakh z vykorystannyam synkhronizuyuchykh poslidovnostey / [M. A. Miroshnik, O. S. Shkil, Y. V. Pakhomov et al.] // Radioelektronika ta informatyka: nauk.-tekhn. zhurnal. – Kharkiv: KHNURE, 2018. – № 3 – S. 82–89 (Indexed by Google Scholar, CiteFactor, NBUV, I2OR, Index Copernicus (<http://journals.indexcopernicus.com/-p24787015.3.html>), SIS, OECSP, OAJI.net, Cyberleninka, Scholar Steer, TIU Hannover).

3. *Pakhomov Y. V.* Design automation of easy-tested digital finite state machines / [M. A. Miroshnik, Y. V. Pakhomov, A. S. Shkil et al.] // Radio Electronics, Computer Science, Control, the scientific journal, Zaporizhzhia National Technical University. – 2018. – № 2. – C. 117–124 (Indexed by Thomson Reuters Web of Science (WoS), CiteFactor, COPAC, CrossRef, eLibrary.ru / РИИЦ, GENERAL IMPACT FACTOR, Google Scholar, Impactfactor.pl, Index Copernicus, Scholar Steer, SIS, SSM).

4. *Pakhomov Y. V.* Analiz aparaturnykh vytrat pri testoprydatnomu proektuvanni keruyuchykh tsyfrovyykh avtomativ / [O. S. Shkil, M. A. Miroshnik, Y. V. Pakhomov, et al.] // Visnyk SNU im. V. Dalya. – 2018. – №6. – S. 101–109. (Indexed by eLibrary.ru, NBUV, WorldWideScience.org, ScienceDirect, Google Scholar, WorldCat, BASE, DOAJ, URAN).

5. *Pakhomov Y. V.* Model of automated hardware diagnostics of remote energy systems management points / M. A. Miroshnik, Y. V. Pakhomov // Svitlotekhnika ta elektroenerhetyka: mizhnar. nauk.-tekhn. zhurnal. – Kharkiv:

KHNUMG im. O. M. Beketova, 2017. – № 3. – S. 3–9 (Indexed by Google Scholar, Index Copernicus, WorldCat).

6. *Pakhomov Y. V.* Metody proyektirovaniya samoproveryayemykh tsifrovyykh avtomatov / [M. A. Miroshnik, E. N. Kulak, Y. V. Pakhomov et al.] // Radiotekhnika: vseukr. mezhved. nauch.-tekhn. sb. – 2016. – №187. – S. 124–131. (Indexed by Google Scholar, HBYB, Elibrary.ru).

7. *Pakhomov Y. V.* Metody sinteza legkotestiruyemykh tsifrovyykh avtomatov / M. A. Miroshnik, Y. V. Pakhomov, A. S. Grebenyuk, I. V. Filippenko // *Ínformatsiyno-keruyuchi sistemy na zalíznichnomu transporti: nauk.-tekhn. zhurnal* – 2016. – №5. – S. 28–39. (Indexed by Google Scholar, PIHЦ, Elibrary.ru).

Results that confirm the approbation of the thesis:

8. *Pakhomov Y.* Design of Logical Control Units Based on Finite State Machines' Patterns / [M. Miroshnyk, Y. Pakhomov, A. Shkil et al.] // IEEE East-West Design & Test Symposium (EWDTS'2018, Kazan, Russia, 14-17 Sept. 2018) – 6 p (Indexed by Scopus).

9. *Pakhomov Y. V.* Microwave multiport multimeter sensor mutual reflection and its influence on signal and tract parameter measurement accuracy / [M. A. Miroshnik, O. B. Zaichenko, Y. V. Pakhomov et al.] // 28-th International Scientific Symposium "Metrology and Metrology Assurance" (Sozopol, Bulgaria, 10–14 Sept. 2018). – C. 60–64 (Indexed by Elibrari.ru, Erih plus, Scince Index, DOAJ, PIHЦ).

10. *Pakhomov Y. V.* Proektirovaniye logicheskikh blokov upravleniya s pomoshch'yu shablonov opisaniya konechnyykh avtomatov / M. A. Miroshnik, Y. N. Salfetnykova, Y. V. Pakhomov. // *Informatsiyno-keruyuchi sistemy na zaliznychnomu transporti: 31-a mizhnarodna nauk.- prakt. konf., (Kharkiv, UkrDUZT, 24–26 zhovt. 2018r.).* – № 4. – S. 21–22 (Indexed by Google Scholar, PIHЦ, Elibrary.ru).

11. *Pakhomov Y.* Design automation of testable finite state machines / [M. Miroshnyk, Y. Pakhomov, S. Shkil et al.] // IEEE East-West Design & Test Symposium (EWDTS'2017, Novi Sad, Serbia, Sept. 27–Oct. 2, 2017). – P 203-208 (Indexed by Scopus, IEEE Xplore).

12. *Pakhomov Y. V.* Issledovanye metodov synteza lehkotestiruemykh tsyfrovyykh ustroystv i system / M. A. Myroshnik, V. A. Krylova, Y. V. Pakhomov, A. N. Miroshnik // Problemy informatyky ta modelyuvannya (PIM-2017): 17-ta mizhnar. nauk.- tekhnich. konf. (Kharkiv, NTU «KHPI», 11–15 ver. 2017r.) S. 58–59 (Indexed by Google Scholar, Index Copernicus, OAJI, Cyberleninka, WorldCat, BASE, UIF, DOAJ, NBUV MIAR, DOI, CrossRef).

13. *Pakhomov Y. V.* Metody obnaruzheniya oshybok proektirovaniya v konechnykh avtomatakh s ispol'zovaniem sinkhroniziruyuschykh posledovatel'nostey / M. A. Myroshnyk, Y. V. Pakhomov // Informatsiyno-keruyuchi systemy na zaliznychnomu transporti: 30- ta mizhnar. nauk.- prakt. konf. (Kharkiv, UkrDUZT, 26–27 zhovt. 2017r.). – № 4. – S. 15–16 (Indexed by Google Scholar, PIHII, Elibrary.ru).

14. *Pakhomov Y. V.* Metody otsinky efektyvnosti remontno-vidnovlyuval'nykh robit dlya vyrobiv hazovogo obladnannya ta truboprovidnykh system / M. A. Miroshnik, V. G. Kotukh, Y. V. Pakhomov. // Informatsiyni tekhnologiyi ta komp'yuterna inzheneriya: 5-ta mizhnar. nauk.- prakt. konf. (Ivano-Frankivs'k, Vinnytsya, 27–29 trav. 2015r.). – S. 15–16 (Indexed by Google Scholar, PIHII, DOAJ).

15. *Pakhomov Y. V.* Strukturnyy analiz izmerytel'nykh system datchykov dlya hazovogo oborudovaniya y truboprovodnykh system / M. A. Myroshnik, V. G. Kotukh, Y. V. Pakhomov. // Informatsiyni tekhnologiyi ta komp'yuterna inzheneriya: 4-ta mizhnar. nauk.- prakt. konf. (Vinnytsya, 28–30 trav. 2014r.). – S. 32–34 (Indexed by Google Scholar, PIHII, DOAJ).

16. *Pakhomov Y. V.* Systemnyy pidkhid do vyznachennya skladu dopomizhnogo materialu dlya vyrobiv gazovogo obladnannya ta truboprovidnykh system / M. A. Miroshnik, V. G. Kotukh, Y. V. Pakhomov // Informatsiyno-

keruyuchi systemy na zaliznichnomu transporti: 27- a mizhnar. nauk.- prakt. konf. (Kharkiv, UkrDUZT, 24-26 ver. 2014r.). – № 4. – S. 30-31 (Indexed by Google Scholar, PIHL, Elibrary.ru).

Publications that additionally reflect the scientific results of the thesis:

17. *Pakhomov Y. V.* Information model of registration and analysis of technological factors arising during final processing of products of transport pipeline systems / V. Kotukh, N. Kaptsova, Y. Pakhomov, V. Kosenko // International Journal of Engineering and Technology (UAE, ISSN: 2227-524X). – 2018. – № 2.23. – P. 73–76 (Indexed by Scopus, IEEE Xplore).

18. *Pakhomov Y. V.* Metody avtomatyzatsiyi proektuvannya lehkotestuemykh komp'yuternykh system i prystroyiv na osnovi tsyfrovyykh avtomativ / M. A. Miroshnik, L. A. Klimenko, Y. V. Pakhomov // Informatsiyno-keruyuchi systemy na zaliznichnomu transporti: naukovo- tekhn. zhurnal. – 2018. – №4 (131). – P. 3–10 (Indexed by Google Scholar, PIHL, Elibrary.ru).

19. *Pakhomov Y. V.* Methods for Designing Self-Checking Digital Machines / [M. Miroshnyk, E. Kulak, Y. Pakhomov ta in.] // Telecommunications and Radio Engineering, USA. – 2017. – № 15. – P. 1367– 1377 (Indexed by Scopus, IEEE Xplore).

20. *Pakhomov Y. V.* Konstruktion of Distributed Information Management Systems for Accounting and Control of Energy Consumption at the Example of Gas / V. Kotukh, Y. Pakhomov // Telecommunications and Radio Engineering, USA. – 2016. – № 7. – P. 631–641 (Indexed by Scopus, IEEE Xplore).

21. *Pakhomov Y. V.* Rozrobka algorytmu prognozuvannya funktsional'no-tekhnichnogo stanu vyrobiv gazovogo obladnannya ta truboprovidnykh system / Y. V. Pakhomov // Visnyk BGTU im. V. G. Shukhova: nauk.- teor. zhurnal. – 2015. – № 3. – P. 90–97 (Indexed by Elibrary.ru, PIHL, IAC, Science index).

22. *Pakhomov Y. V.* Osnovni pryntsypy stvorenniya yedynoyi rozpodileno-yi systemy avtomatichnogo kontrolyu ta obliku energoresursiv na prykladi gazovoyi

galuzi / V. G. Kotukh, Y. V. Pakhomov // Informatsiyno-keruyuchi systemy na zaliznichnomu transporti: naukovo- tekhnichnyy zhurnal. – Kharkiv: UkrDUZT, 2015. – № 3. – P. 48–55 (Indexed by Google Scholar, PIHL, http://elibrary.ru/title_about.asp?id=33934).

23. *Pakhomov Y. V. Pobudova rozpodil'nykh informatsiyno keruyuchykh system obliku i kontrolyu enerhoresursiv na prykladi hazovoyi haluzi / V. G. Kotukh, Y. V. Pakhomov // Radiotekhnika: vseukr. mizhvid. nauk.- tekhn. zb. – Kharkiv: KHNURE, 2015. – № 182. – P. 65–72 (Indexed by Google Scholar, HBYB, eLIBRARY.RU).*

24. *Pakhomov Y. V. Issledovaniye raspredeleniya teplovykh potokov pri impul'snoy lazernoy svarke korpusov datchikov dlya gazovogo oborudovaniya i truboprovodnykh sistem / M. A. Miroshnik, V. G. Kotukh, Y. V. Pakhomov. // Vestnik BGTU im. V. G. Shukhova. – 2014. – №5. – P. 96–101. (Indexed by HEB Elibrary.ru, PIHL, IAC, Science index).*

25. *Pakhomov Y. V. Tekhnologicheskaya kontseptsiya sozdaniya ASU TP dlya ob'yektov energosnabzheniya na baze tsifrovykh raspredelennykh sistem / M. A. Miroshnik, V. G. Kotukh, Y. V. Pakhomov. // Radiotekhnika: vseukr. mezhved. nauch.- tekhn. sb. – 2014. – №179. – P. 131–137 (Indexed by Google Scholar, HBYB, Elibrary.ru).*

26. *Pakhomov Y. V. Metody rascheta uprugikh elementov membranno-balochnogo tipa v datchikakh dlya gazovogo oborudovaniya i truboprovodnykh sistem / Y. V. Pakhomov. // Komunal'ne gospodarstvo mist. – 2014. – №116. – P. 64–66. (Indexed by Google Scholar, WorldCat, DOAJ, HBYB).*

Key words: critical system, control finite state machine, design, diagnosis, testability, hardware costs, hardware description languages, CAD FPGA.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	32
ВСТУП.....	33
1 АНАЛІЗ МЕТОДІВ ПРОЕКТУВАННЯ ТА ДІАГНОСТУВАННЯ КЕРУЮЧИХ АВТОМАТІВ В СИСТЕМАХ ЛОГІЧНОГО УПРАВЛІННЯ	41
1.1 Управління в системах критичного застосування.....	41
1.2 Автоматні моделі в критичних системах логічного управління...	45
1.3 Методи діагностування автоматних моделей	52
1.4 Методи тестопридатного проектування цифрових пристроїв	57
1.5 Постановка мети і завдань дослідження	64
2 МОДЕЛІ ТА МЕТОДИ ДІАГНОСТУВАННЯ ЦИФРОВИХ КЕРУЮЧИХ АВТОМАТІВ	67
2.1 Моделі автоматного пристрою керування ГРП	67
2.2 Діагностичні експерименти над керуючими автоматами.....	80
2.3 Апаратне діагностування керуючих автоматів.....	87
2.4 Висновки до розділу 2	96
3 ТЕСТОПРИДАТНЕ ПРОЕКТУВАННЯ КЕРУЮЧИХ АВТОМАТІВ..	97
3.1 Побудова легкотестованих автоматів шляхом розширення вхідного алфавіту	98
3.2 Проектування HDL-моделей тестопридатних автоматів	103
3.2.1 Тестопридатна структура з використанням регістру зсувну	107
3.2.2 Тестопридатна структура із скануванням станів.....	109
3.3 Аналіз апаратурних витрат при побудові легкотестованих автоматів.....	116
3.3.1 Аналіз тестопридатності функцій переходів.....	116
3.3.2 Розрахунок тестопридатності графових моделей автоматів..	127
3.4 Процедури розміщення додаткових дуг у графі переходів.....	131
3.5 Висновки до розділу 3.....	141

4 АВТОМАТИЗАЦІЯ ПРОЕКТУВАННЯ ТЕСТОПРИДАТНИХ АВТОМАТІВ.....	142
4.1 Тестопридатне проектування керуючого автомату.....	142
4.2 Програмний модуль побудови тестопридатних HDL-моделей.....	148
4.3 Тестопридатне проектування системи управління ГРП.....	154
4.4 Висновки до розділу 4.....	158
ВИСНОВОК.....	160
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	162
ДОДАТОК А. СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ.....	178
ДОДАТОК Б. ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ.....	184
ДОДАТОК В. ДОКУМЕНТИ, ЩО ПІДТВЕРДЖУЮТЬ ВПРОВАДЖЕННЯ.....	185
ДОДАТОК Г. VHDL-МОДЕЛІ КА КП ГРП, СИСТЕМА ВЕРИФІКАЦІЇ ТА ДІАГНОСТУВАННЯ.....	190
ДОДАТОК Д. МОДЕЛЬ ПОБУДОВИ СИНХРОНІЗУЮЧИХ ПОСЛІДОВНОСТЕЙ ДЛЯ КА КП ГРП.....	196
ДОДАТОК Е. VHDL-МОДЕЛІ АПАРАТНОГО ПРИСТРОЮ ДІАГНОСТУВАННЯ.....	198
ДОДАТОК Ж. РЕЗУЛЬТАТИ АВТОМАТИЗОВАНОГО СИНТЕЗУ VHDL-МОДЕЛЕЙ КА МПА СКЛАДАННЯ.....	211
ДОДАТОК К. РЕЗУЛЬТАТИ АВТОМАТИЗОВАНОЇ ПОБУДОВИ VHDL-МОДЕЛЕЙ КА КП ГРП.....	213

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АС	–	автоматизовані системи;
ГСА	–	граф-схеми алгоритмів;
ГРП		газорегуляторний пункт;
ДЕ	–	діагностичний експеримент;
ДП	–	дискретний пристрій;
КА	–	керуючий автомат;
МОА		мови опису апаратури;
МПА		мікропрограмний автомат
ОД	–	об'єкт діагностування;
ПЗ	–	програмне забезпечення;
ПЛІС	–	програмовані логічні інтегральні схеми;
РЕП	–	радіоелектронні пристрої;
САПР	–	система автоматизованого проектування;
СЛУ	–	система логічного управління;
ТОР	–	технічне обслуговування та ремонт;
ТПВ	–	таблиця переходів-виходів;
ЦП	–	цифровий пристрій;
ASM	–	Algorithmic State Machine;
BIST		Built-in Self-Test, (вбудоване самотестування)
FPGA	–	Field Programmable Gate Array
FSM	–	Finite State Machine, (кінцевий автомат);
HDL	–	Hardware Description Language, (мови опису апаратури);
RTL	–	Register Transfer Level, (рівень регістрових передач);
SoC	–	System-on-Chip, (система на кристалі);
TDI	–	test data input, (вхід подачі тестових впливів);
VHDL	–	Very High Speed Integrated Circuits HDL, (одна з мов опису апаратури).

ВСТУП

Актуальність теми дослідження. Критичні технічні системи, тобто такі, у яких відмови можуть призвести до значних економічних втрат, фізичних руйнувань або загроз життю людей, вимагають особливої уваги щодо надійності їх функціонування. Для підвищення надійності критичних систем, якості технічного обслуговування, продуктивності праці технічного персоналу і своєчасного виявлення відмов, використовуються системи контролю й діагностики.

Апаратні критичні системи, які застосовуються в електроенергетиці та газопостачанні, характеризуються тим, що при проведенні регламентних робіт щодо технічного обслуговування їх зупинка на тривалий час для проведення діагностування неможлива, а функціональне діагностування є дуже витратним. Виходячи з цього для діагностування апаратних критичних систем використовуються вбудовані системи контролю та діагностики.

У критичних системах енерго- та газопостачання широко використовуються спеціальні локальні системи логічного управління й регулювання на основі композиції керуючого та операційного автоматів, які розташовані на технологічно відокремлених об'єктах. Для контролю автоматних систем логічного управління доцільно використовувати апаратні системи, які імітують алгоритм управління чи діагностування, або методи тестопридатного проектування, які перетворюють керуючі автомати на легкотестовані, у яких витрати на проведення діагностичного експерименту є мінімальними за довжиною та часом. При проектуванні систем логічного управління на технологічній платформі ПЛІС з використанням САПР на основі мов опису апаратури вбудовані системи діагностування або апаратні засоби підвищення тестопридатності доцільно вводити ще на етапі проектування, тобто на етапі побудови моделей мовами опису апаратури.

Таким чином, актуальним і важливим науко-технічним завданням є розробка моделей і методів автоматизованого проектування та

діагностування тестопридатних цифрових систем логічного управління на основі кінцевих автоматів.

Значний внесок у вирішення проблем тестопридатного проектування та діагностування цифрових систем на основі кінцевих автоматів внесли провідні вітчизняні й зарубіжні вчені: А.П. Горяшко, Є.С. Согомоян, В.Г. Тоценко, Д.В. Сперанський, Л.В. Дербунович, М.Л. Малиновський, В.В. Соловйов, О.А. Баркалов, В.С. Харченко, Р-Й. Убар, Г.Ф. Кривуля, В.І. Хаханов, E.G. Hennie, P.G. Bennetts, D. Harel, M. Breuer, M. Abramovici, J. Roth, Y. Zorian, D.Hanna та інші.

Зв'язок роботи з науковими програмами та темами. Розробка розділів дисертації здійснювалася відповідно до планів держбюджетних НДР і договорів, що виконувалися на кафедрі автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки – держбюджетна НДР «Розумний Кібер Університет – Cloud-Mobile – сервіси управління науково-освітніми процесами», (номер державної реєстрації 0117U0002524 (2017–2018)) та кафедри спеціалізованих комп'ютерних систем Українського державного університету залізничного транспорту – держбюджетна НДР «Формування теоретичних засад підвищення ефективності використання інформаційно-керуючих систем на залізничному транспорті» № 23/1-2016Б (2015–2017). Здобувач брав участь у проведенні зазначених робіт як виконавець і розробив методи, алгоритми та програмні засоби тестопридатного проектування систем керування на основі кінцевих автоматів.

Науково-практична задача дослідження – розробка моделей і методів автоматизованого проектування та діагностування тестопридатних цифрових систем логічного управління на основі кінцевих автоматів.

Сутність дослідження полягає в підвищенні надійності функціонування критичних систем логічного управління за рахунок скорочення часу та підвищення якості методів діагностування у рамках проведення регламентних робіт щодо технічного обслуговування апаратної

частини локальних систем управління технологічно відокремлених об'єктів електроенергетики та газопостачання, побудованих на основі кінцевих цифрових автоматів. Скорочення часу та підвищення якості методів діагностування досягається за рахунок розробки методів проведення тестопридатного проектування та діагностування автоматних систем логічного управління, побудованих на технологічній платформі програмованих логічних інтегральних схем. Тестопридатне проектування здійснюється з використанням інструментальних засобів систем автоматизованого проектування на основі мов опису апаратури.

Ринкова привабливість дослідження. Впровадження моделей та методів тестопридатного проектування критичних систем логічного управління на основі керуючих автоматів приведе до зменшення часу на проектування та технічне обслуговування критичних автоматних систем логічного управління та кількості обслуговуючого персоналу.

Об'єкт дослідження – процес забезпечення тестопридатності цифрових систем на основі кінцевих автоматів.

Предмет дослідження – моделі та методи тестопридатного проектування і діагностування автоматних систем логічного управління на ПЛІС.

Мета дослідження – розробка моделей і методів автоматизованого проектування та діагностування автоматних систем логічного управління на ПЛІС з використанням мов опису апаратури для скорочення часу технічного обслуговування критичних систем.

Задачі дослідження:

- 1) визначити особливості діагностування критичних апаратних систем логічного управління;
- 2) удосконалити методи неруйнівних діагностичних експериментів для моделей кінцевих автоматів;
- 3) розробити методи побудови апаратних систем підтримки проведення діагностичних експериментів в автоматних системах керування;

4) удосконалити методи оцінки тестопридатності графових моделей кінцевих автоматів;

5) розробити методи підвищення тестопридатності моделей кінцевих автоматів за рахунок розширення вхідного алфавіту;

6) реалізувати розроблені моделі в програмних модулях побудови тестопридатних автоматів в САПР ПЛІС.

Наукова новизна:

1) вперше запропоновано розширення вхідного алфавіту кінцевого автомата, яке характеризується введенням додаткового стовпця в таблицю переходів-виходів автомата, що суттєво скоротило довжину та час діагностичного експерименту за рахунок можливості встановлювати автомат в довільний стан;

2) отримали подальший розвиток методи проведення діагностичного експерименту за рахунок використання синхронізуючих послідовностей, що дозволяє проводити неруйнівний діагностичний експеримент навіть для автоматів з несправностями функцій переходів;

3) отримали подальший розвиток методи побудови апаратних пристроїв діагностування за рахунок реалізації стратегії обходу всіх станів або дуг графа переходів керуючого автомата, що дозволяє проводити діагностичні експерименти, не порушуючи основний режим роботи критичної системи логічного управління протягом тривалого часу;

4) удосконалено методи розрахунку тестопридатності кінцевих керуючих автоматів, які, на відміну від існуючих методів, дозволили мінімізувати додаткові апаратні витрати при побудові легкотестованих автоматів.

Практичне значення отриманих результатів досліджень полягає у розробці методики, що дозволяє будувати тестопридатні моделі кінцевих автоматів з використанням мов опису апаратури САПР РЕП та шаблонів автоматного програмування. Розроблені процедури розрахунку тестопридатності кінцевих керуючих автоматів дали можливість

оптимізувати створення додаткових переходів у графових моделях автоматів, що дозволило зменшити апаратні витрати при побудові легкотестованих керуючих автоматів. Розроблено програмний модуль із візуальним інтерфейсом для автоматизованого проектування тестопридатних HDL-моделей кінцевих керуючих автоматів. Програмний модуль можна реалізувати у вигляді хмарного сервісу.

Практична реалізація моделей і методів тестопридатного проектування критичних систем логічного управління на основі кінцевих автоматів виконана в рамках створення й верифікації програмних компонентів із візуальним інтерфейсом для автоматизованого проектування тестопридатних HDL-моделей кінцевих керуючих автоматів.

Результати дисертації у складі моделей, методів і фрагментів програмних застосунків упроваджено в навчальному процесі Харківського національного університету радіоелектроніки (акт про впровадження від 21.12.2018).

Розроблено HDL-моделі, які моделюються та синтезуються інструментальними засобами систем автоматизованого проектування цифрових пристроїв на технологічній платформі ПЛІС. Розроблено програмний модуль із візуальним інтерфейсом для вводу графа переходів керуючого автомата для автоматизованої побудови тестопридатних HDL-моделей автоматів у формі автоматного шаблону в синтезованій підмножині мови опису апаратури VHDL, який був застосований у підрозділах ПрАТ «Енергооблік». Це підприємство є розробником мікропроцесорних вимірювальних приладів та систем тестопридатного проектування (довідка про впровадження від 28.11.2018).

Розроблено методики апаратного діагностування локальних пристроїв керування та регулювання технологічно відокремлених об'єктів електроенергетики та газопостачання. Алгоритми керування описуються граф-схемою алгоритму, яка представляється математичною моделлю графа переходів кінцевого автомата. Для реалізації алгоритму діагностування

використовується спосіб обходу всіх дуг графа та моделі керуючого автомата й пристрою діагностування, які описуються мовою опису апаратури VHDL з наступним синтезом та використанням пристроїв програмованої логіки. Ці пристрої були застосовані в підрозділах АТ «Укртрансгаз» для створення систем контролю на автоматичних газорозподільчих станціях, обладнаних сучасними автоматичними приладами, розробленими на базі цифрової техніки (довідка про впровадження від 25.10.2018).

Результати дисертаційної роботи у вигляді автоматної моделі апаратної системи діагностування цифрових пристроїв управління в електроенергетиці й газопостачанні також було застосовано в діяльності підрозділів ПАТ «Харківміськгаз» для створення надійних систем контролю в міських автоматизованих газорозподільчих пунктах, які оснащені сучасним автоматичним обладнанням на базі цифрової техніки. Це дозволило суттєво скоротити час на виявлення та усунення можливих дефектів, які виникають на об'єктах газотранспортної системи, та підвищити ступінь надійності транспортування газу. Впровадження результатів дослідження у виробничу діяльність ПАТ «Харківміськгаз» (довідка про впровадження від 15.11.2018) забезпечило ефективність вимірів технічних параметрів роботи технологічного обладнання ГРП та вузлів обліку газу, що дозволило аналізувати залежність зміни одних параметрів від інших і, як наслідок, підвищення надійності газопостачання, а також спостереження тенденції щодо скорочення комерційних втрат газу.

Особистий внесок здобувача. Усі наукові й практичні результати отримані автором особисто. У роботах, опублікованих у співавторстві, здобувачеві належать: [1] – розробка технологічної частини проекту ремонтно-механічного підприємства для газових і енергетичних господарств; [2] – процедури проведення діагностичного експерименту з використанням синхронізуючих послідовностей; [3] – синтез розширених HDL-моделей керуючого автомата Мура інструментальними засобами автоматизованого проектування; [4] – методи зменшення апаратних витрат під час

проектування тестопридатних кінцевих автоматів за рахунок оптимального розміщення додаткових дуг у графі переходів; [5] – методи побудови апаратних пристроїв діагностування, що реалізують стратегію обходу всіх станів або дуг графа переходів керуючого автомата; [6] – метод проектування цифрових автоматів, що самоперевіряються, і синтез схем вбудованого тестового контролю з використанням методів компактного тестування; [7] – використання шаблонів автоматного програмування під час опису систем логічного управління в САПР РЕП; [8] – використання шаблонів автоматного програмування під час опису систем логічного управління в САПР РЕП; [9] – взаємне відображення датчика багатопортового мультиметра і його вплив на точність вимірювання параметрів сигналу й тракту; [10] – проектування логічних блоків управління за допомогою шаблонів опису кінцевих автоматів; [11] – метод розширення таблиці переходів-виходів автомата, який забезпечує режим обходу всіх вершин графа переходів автомата (станів) в режимі діагностування; [12] – синтез легкотестованих цифрових пристроїв і систем; [13] – метод виявлення помилок проектування в кінцевих автоматах із використанням синхронізуючих послідовностей; [14] – методи оцінки ефективності ремонтно-відновлювальних робіт для газового обладнання; [15] – структурний аналіз вимірювальних систем датчиків для газового обладнання та трубопровідних систем; [16] – методи визначення складу допоміжного матеріалу для виробів газового обладнання; [17] – інформаційна модель обліку й аналізу технологічних факторів; [18] – методи автоматизації проектування легкотестованих комп'ютерних систем і пристроїв на основі цифрових автоматів; [19] – методи проектування легкотестованих цифрових машин; [20] – дослідження інформаційних систем обліку та контролю енергоспоживання; [21] – розробка алгоритму прогнозування функціонально-технічного стану виробів газового обладнання; [22] – принципи створення єдиної розподіленої системи автоматичного контролю та обліку енергоресурсів; [23] – побудова розподілених інформаційно-керуючих систем обліку й контролю

енергоресурсів; [24] – датчики для газового обладнання та трубопровідних систем; [25] – концепція створення АСУ ТП для об'єктів енергопостачання на базі цифрових розподілених систем; [26] – датчики для газового обладнання та трубопровідних систем.

Апробація результатів дисертації. Результати роботи були представлені та обговорені на наступних конференціях: Міжнародна науково-практична інтернет-конференція «Енергозберігаючі технології теплогазопостачання, будівництва та муніципальної інфраструктури» (Харків, ХНУМГ ім. О. М. Бекетова, 2013); IV та V Міжнародні науково-практичні конференції «Інформаційні технології та комп'ютерна інженерія» (Івано-Франківськ, Вінниця, 2014–2015); Міжнародна науково-практична конференція «Інформаційно-керуючі системи на залізничному транспорті» (Харків, УкрДУЗТ, 27-а, 2014; 30-а, 2017; 31-а, 2018); IEEE East-West Design & Test Symposium (EWDTS'2017, Novi Sad, Serbia, Sept. 27–Oct. 2, 2017, та EWDTS'2018, Kazan, Russia, 14-17 Sept. 2018); XVII Міжнародна науково-технічна конференція «Проблеми інформатики та моделювання (Харків, НТУ «ХПИ», 2017); 28-th International Scientific Symposium "Metrology and Metrology Assurance" (Sozopol, Bulgaria, 2018).

Публікації. Результати дисертаційної роботи відображено у 26 друкованих працях, серед яких 1 монографія, 13 статей у наукових журналах із Переліку наукових фахових видань України, 3 статті в міжнародних наукових журналах за кордоном, 5 статей входять до міжнародної наукометричної бази Scopus; 1 стаття – Web of Science; 9 матеріалів міжнародних наукових конференцій (з них 3 за кордоном і 2 входять до наукометричної бази Scopus). Здобувачеві належать 6 публікацій, що входять до наукометричної бази Scopus, індекс Хірша $h=1$.

Структура і обсяг дисертації. Дисертація представлена на 215 сторінках (із них 133 сторінки основного тексту) і містить 4 розділи, 77 рисунків, 5 таблиць, список джерел із 140 найменувань (на 16 с.), 8 додатків (на 37 с.), анотації на 28с.

РОЗДІЛ 1

АНАЛІЗ МЕТОДІВ ПРОЕКТУВАННЯ ТА ДІАГНОСТУВАННЯ КЕРУЮЧИХ АВТОМАТІВ В СИСТЕМАХ ЛОГІЧНОГО УПРАВЛІННЯ

1.1 Управління в системах критичного застосування

У сучасному суспільстві промислове виробництво грає важливу роль по задоволенню матеріальних і духовних потреб, що спричиняє за собою збільшення масштабів виробництва. Внаслідок чого промислове виробництво стало постійним джерелом виникнення нещасних випадків, аварій і катастроф. Найбільшою мірою аварійність властива вугільній, гірничорудній, хімічній, нафтогазовій і металургійній галузям промисловості, транспорту. Проблема попередження аварій набуває особливу актуальність в атомній енергетиці, хімічній промисловості, при експлуатації військової техніки, де використовуються і звертаються потужні джерела енергії, високотоксичні і агресивні речовини. [1, 2].

Міжнародні стандарти та державні стандарти України визначають відмови технічних систем (елементів, технологічних процесів) як причина аварійних ситуацій [3, 4].

Вид відмови – сукупність можливих відмов елементу і/або системи, об'єднаних в деяку класифікаційну групу по спільності одного або декількох ознак (причини, механізм виникнення, зовнішні прояви і інші ознаки, окрім наслідків відмови). Тяжкість наслідків відмови - якісна або кількісна оцінка вірогідного (спостережуваного) збитку від відмови елементу і/або системи. Критична відмова – відмова системи або її елементу, тяжкість наслідків якої в межах цього аналізу визнана неприпустимою і вимагає вживання спеціальних заходів по зниженню ймовірності цієї відмови і/або можливого збитку, пов'язаного з її виникненням.

Таким чином, критичні технічні системи - це системи, в яких відмови можуть привести до значних економічних втрат, фізичних пошкоджень або

погроз людському життю. До найбільш тяжких наслідків в відмовах критичних систем призводять відмови в системах управління. Вимоги, що пред'являються до систем управління в критичних системах в частині забезпечення безпеки стають усе більш жорсткими. При цьому з розвитком комп'ютерних технологій все більше відповідальних функцій перекладається з людини на обчислювальну техніку. У цих умовах проблема підвищення безпеки систем управління набуває все більшого значення [5].

Для визначення поняття безпеки уся множина станів S розбивається на підмножини S_c - справних станів (відмови відсутні), S_p - працездатних станів (є один або декілька відмов, які не призводять до зміни (погіршенню) значень параметрів системи за рахунок наявності засобів надмірності), S_z - захисних станів і S_n - небезпечних станів. Виходячи з цього, безпека - це властивість системи безперервно зберігати справний, працездатний або захисний стан протягом деякого часу або напрацювання [6–8].

Завдання забезпечення безпеки можна розділити на забезпечення безпеки апаратних (hardware) і програмних (software) засобів. У свою чергу, безпека апаратних засобів визначається безпекою безпосередньо апаратури і систем обробки інформації, а також безпекою пристроїв узгодження систем управління із виконуваними об'єктами.

Згідно [9] виділяються вісім класів безпечних апаратних систем. У системах шостого класу з кодованими змінними безпека забезпечується роботою вихідного перетворювача (дешифратора), який у разі відмов або збоїв переводить систему в захисний стан. Принцип квазібезпеки без кодування інформації (сьомий клас) широко застосовується на макрорівні в мікропроцесорних системах. Схеми восьмого класу (з надлишковими контрольними засобами і кодуванням логічних змінних), як правило, застосовуються в системах передачі відповідальної інформації, у тому числі телекерування і телесигналізації.

При розробці надійних дискретних систем автоматичного і автоматизованого управління транспортом і промисловістю

використовуються схеми що самоперевіряються, а також схеми вбудованого і надбудованого контролю [10–12]. Методика їх побудови базується на теорії системного функціонального (робочого) діагностування - такого способу визначення технічного стану об'єкта діагностування, при якому він не відключається від керованих об'єктів, а всі його вхідні сигнали є одночасно і тестовими [13]. Для організації перевірки правильності функціонування цифрового блоку і, тим самим, непрямого визначення його технічного стану, він забезпечується спеціалізованою схемою контролю [14]. Схема контролю може бути реалізована за методом або вирахуванням контрольних розрядів [15], або логічним доповненням [16].

Одними з найбільш аварійно небезпечних критичних систем є системи газопостачання. Для забезпечення їх безперервного функціонування створена автоматизована система оперативно диспетчерського управління (АСОДУ), яка являє собою багаторівневу програмно-апаратну систему, яка забезпечує надійне функціонування системи газопостачання на різних рівнях ієрархії [17], яка представлена на рис. 1.1.

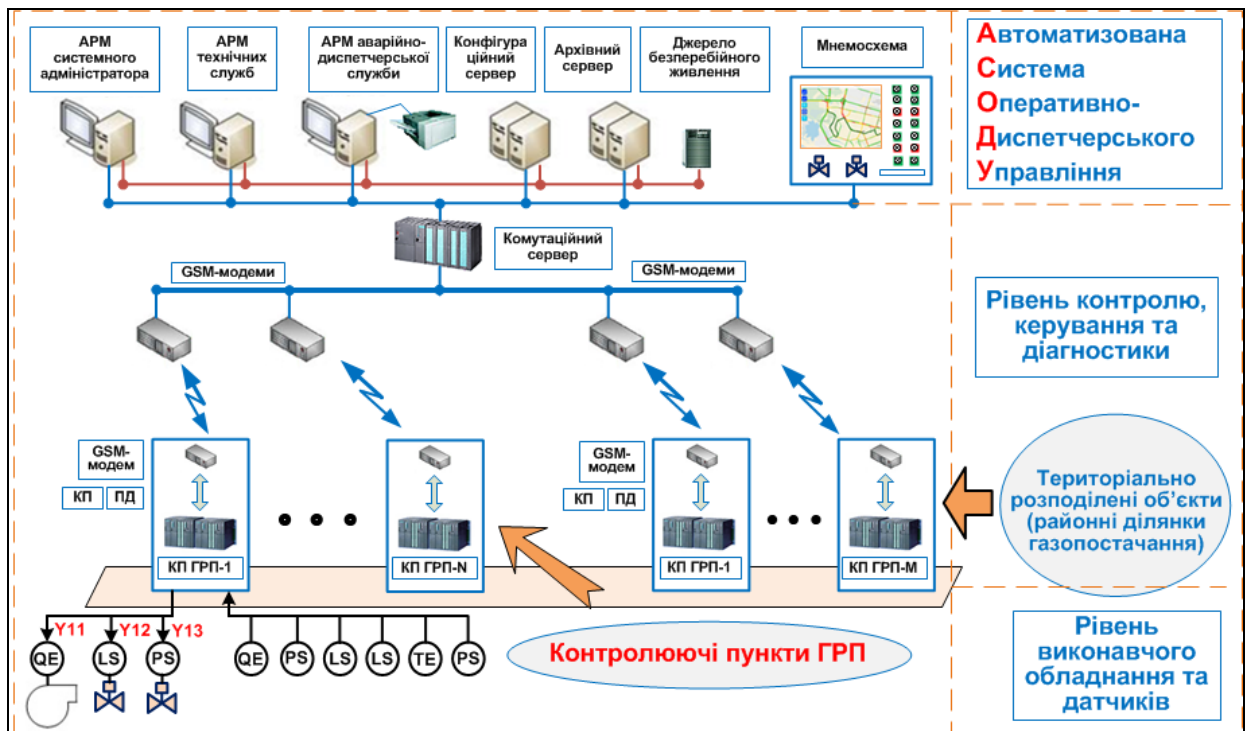


Рисунок 1.1 – Загальна структура АСОДУ у системі газопостачання

На нижньому рівні цієї системи функціонують локальні системи управління газорозподільчими станціями (ГРС) та газорегуляторними пунктами (ГРП), які функціонують, як правило, у режимі територіально віддалених об'єктів [18]. Одним із шляхів досягнення високої надійності критичних технічних систем у газопостачанні є проведення регламентних робіт з їх технічного обслуговування та діагностування. При цьому застосовуються різні технології неруйнівного контролю та діагностування, в тому числі діагностування локальних систем управління ГРП. Особливістю технічного діагностування систем управління в газопостачанні є те, що для проведення технічного діагностування зазначені системи не можна зупинити, а тим більше демонтувати [19]. Сучасним трендом в розвитку систем безпеки в газопостачанні є все більш широке використання цифрових технологій [20].

Нині проблема підвищення безпеки програмного забезпечення (ПЗ) є надзвичайно актуальною, про що свідчить велика кількість наукових робіт, що ведуться в цьому напрямі, а також інструментальних засобів, що розробляються з метою вдосконалення технології проектування ПЗ для систем управління критичними системами. Нині безпека ПЗ, як правило, досягається за рахунок застосування часової, інформаційної і структурної надлишковості, а також спеціальних методів тестування ПЗ. [21],

У [22] показані особливості застосування програмних систем критичного застосування для різних прикладних областей: аерокосмічна техніка, атомній енергетика тощо. У [23] розглянуті питання розробки ПЗ для критичних безпечних систем, способів обробки ризиків, критичних для безпеки систем, їх специфікацій та стандартів, мовної підтримки та підходів до проектування таких систем. У [24] викладені загальні принципи проектування ПЗ критичних систем та вимоги до їх експлуатації. При цьому важливу роль грає комплексне тестування програмних систем [25].

1.2 Автоматні моделі в критичних системах логічного управління

Основною моделлю при побудові цифрових систем переробки інформації та управління є кінцевий автомат (Finite State Machine, FSM). Кінцевим автоматом називається математична модель синхронної системи для перетворення вхідної інформації, представлені кінцевим алфавітом у вихідну інформацію, представлену також кінцевим алфавітом через кінцеве число стійких станів [26]. Для схемної реалізації кінцевих автоматів використовується модель структурного автомата, яка є композицією комбінаційної частини, що реалізує функції переходів та виходів, і запам'ятовуючої частини, яка забезпечує збереження станів автомата [27]. Структурний автомат в пристроях обробки інформації або управління традиційно представляється композицією операційного автомата (ОА) та керуючого автомата (КА) [28].

Одним з класів систем управління є системи логічного управління (СЛУ). Специфіка СЛУ полягає в тому, що у них вхідні сигнали X і вихідні сигнали Y можуть приймати тільки два значення - 0 та 1. Об'єктами керування (ОК) для систем логічного управління є клапани, вентилятори, електродвигуни, пристрої освітлення, пристрої регулювання температури, вологості, тиску тощо. СЛУ впливають на ОК не безпосередньо, а через виконавчі механізми - магнітні пускачі, електромеханічні перетворювачі, гідроромеханічні перетворювачі, регулятори тиску, терморегулятори тощо. Тому при побудові таких систем передбачається, що вони входять до складу ОК. Інформація про роботу ОК надходить в СЛУ від сигналізаторів положення, стану і параметрів. При цьому необхідно зазначити, що сигналізатори параметрів складаються з двох складових - датчика і вторинного приладу, що здійснює порівняння значення аналогового сигналу від датчика із заданими значеннями параметрів, і формування двійкового сигналу $\{0, 1\}$ при виході значення за межі встановлених діапазонів [29].

При використанні автоматних моделей для опису систем і компонентів

критичного застосування, слід враховувати особливості, які характерні для цих систем. Найчастіше алгоритми логічного управління, що розробляються, та архітектура систем логічного управління вступають в конфлікт, який загострюється зі збільшенням складності алгоритмів і підвищення вимог що пред'являються до розроблюваних систем управління. Особливу важливість при їх розробці набувають такі завдання, як точна і коректна інтерпретація вимог замовника, створення тестопридатних архітектур керуючих автоматів, проектування безпечного програмного забезпечення для них, безпечне узгодження мікроелектронних структур керуючих автоматів з виконавчими об'єктами тощо.

Особливості функціонування автоматних систем логічного управління критичного застосування реалізуються за рахунок багатоверсійних систем із загальною і роздільною диверсностью [30], за рахунок використання автоматних моделей безпечних контролерів на програмованих логічних інтегральних схемах (ПЛІС-контролерів) [31], за рахунок процедури абстрактного і структурного синтезу безпечних автоматів по формальному опису вимог до безпеки, заснованого на формуванні множин відповідальних функцій [32], за рахунок спеціальних видів кодування станів з урахуванням технологічної платформи апаратної реалізації [33, 34, 35], за рахунок проектування строго безпечних автоматів, що самоперевіряються [36], за рахунок сигнатурного моніторингу функцій виходів автомата [37] тощо.

Ще одним напрямком в проектуванні автоматних критичних систем є зниження енергоспоживання. В [38] при проектуванні кінцевих автоматів на ПЛІС FPGA запропонований підхід, який базується на обмеженні перемикання переходів між послідовними станами FSM. Результати розробки показали значне зниження на 60,55% і 66,86% в динамічному споживанні енергії. У [39] розглянуті структурні моделі кінцевих автоматів, які дозволяють використовувати значення вихідних змінних як коди внутрішніх станів. Показано, що запропонований метод дозволяє понизити вартість реалізації та енергоспоживання КА для усіх сімейств ПЛІС різних

виробників в середньому в 1.41-1.72 разу, а для окремих сімейств - в 2 рази.

Створення будь-якої системи управління починається з рівня найвищої абстракції, коли в найзагальнішому і неформальному вигляді, формулюються вимоги до виконуваних системою функцій. Всі наступні етапи розробки супроводжуються неминучим проявом протиріч між вимогами замовника і технічними можливостями системи управління об'єктами критичного застосування, які мають при цьому свої особливості.

Проектування автоматних систем логічного управління (в частині керуючого автомата) складається з наступних етапів [40]:

- визначення технічних станів об'єкта управління;
- визначення вхідних та вихідних сигналів для кожного стану;
- обрання типу моделі керуючого автомата (Мілі, Мура);
- визначення кількості станів керуючого автомата;
- формування логічних виразів функцій переходів та виходів;
- побудова змістовного графа переходів керуючого автомата;
- перевірка побудованого графа переходів на синтаксичну та семантичну коректність.

Таки чином, в якості алгоритмічної моделі для КА найчастіше застосовується такий непроцедурний візуальний формалізм, як графи переходів, які в англійській літературі мають назву діаграми станів (State Diagrams). Зазначимо, що змістовні графи переходів КА є не тільки візуальним відображенням алгоритму функціонування автомата, але й повною його математичною моделлю [26, 41].

При існуючому різноманітті вхідних форм опису проектів цифрових пристроїв (ЦП) можна виділити найбільш популярні, а саме: аналітичні - мови опису апаратури (Hardware Description Language HDL), графічні або візуальні - ієрархічні цифрові структури і схеми, граф-схеми алгоритмів (ГСА) операційних або керуючих пристроїв (Algorithmic State Machine, ASM). Одним з поширених способів вхідного опису кінцевого керуючого автомата на мові опису апаратури є автоматний шаблон, тобто спеціальна

структура HDL-коду, яка будується на основі графа переходів автомата або прямої структурної таблиці (ПСТ). Побудова графа переходів кінцевого автомата на основі інших способів опису його функціонування є мистецтвом проектувальника і особливостями систем вводу проектів інструментальних засобів систем автоматизованого проектування радіоелектронної апаратури (САПР РЕА) [42, 43].

При цьому важливу роль грають візуальні засоби вводу описів кінцевих автоматів. До засобів візуального програмування відносяться графічні редактори і редактори станів та редактори часових діаграм. Графічні редактори дозволяють наочно об'єднувати окремі функціональні вузли в єдине цифровий пристрій. Область ефективного застосування редакторів тимчасових діаграм обмежується описом нескладних комбінаційних схем. У теперішній час існує багато програмних продуктів, які мають візуальні засоби вводу графових моделей, в тому числі on-line продукти, які дозволяють суттєво спростити процес розробки та верифікації автоматних моделей [44].

Одними з найбільш цікавих програмних продуктів з цього переліку є інструментальні засоби компанії HDL Works [45]. Компанія HDL Works розробляє і продає високопродуктивні, інтуїтивно зрозумілі інструменти для проектування і верифікації HDL-моделей та проектів на ПЛІС FPGA в широкому спектрі застосувань. Усі програмні продукти працюють під керуванням ОС Microsoft Windows або Linux. Головний офіс компанії HDL Works знаходиться в м. Ede, Нідерланди. Одним з on-line продуктів компанії HDL Works є State diagram editor. Цей редактор станів КА підтримує автоматні моделі Moore, Mealy і змішаних автоматів. Для визначення дій і умов переходу можна використовувати будь-який оператор VHDL або Verilog. Переходи можуть бути синхронними або асинхронними; виходи можуть бути тактованими або комбінаторними. Редактор підтримує різні методи кодування станів, у тому числі користувачькі. Візуальний інтерфейс продукту HDL Works State diagram editor наведений на рис.1.2.

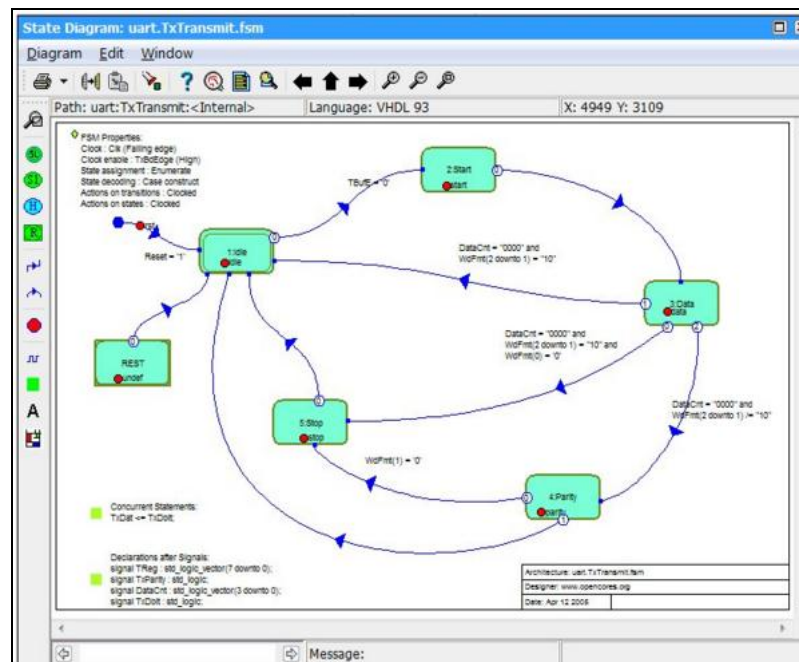


Рисунок 1.2 – Вікно вводу HDL Works State diagram editor

При описі алгоритму функціонування цифрових пристроїв логічного керування в САПР цифрових пристроїв одним із стилів написання коду мовами опису апаратури (HDL-коду) є стиль автоматного програмування. Суть автоматного програмування полягає в відділенні опису логіки поведінки (за яких умов необхідно виконати ті чи інші дії) від опису його семантики (власне сенсу кожного з дій). Автоматні програми строго структуровані і в них виділено три види функцій: функції переходів, функції виходів та функції призначення нового стану. Автоматні програми також строго шаблонізовані з використанням операторів багатопозиційного вибору (switch, case) та умовних операторів (if, elsif-else) [46, 47]. Стиль автоматного програмування на мовах опису апаратури також має враховувати особливості синтезованої підмножини HDL-коду [43].

Сучасний цикл проектування цифрових систем з використанням САПР на основах мов опису апаратури, є складним багатоетапним інженерним процесом [48, 49], і при цьому необхідно зважати на специфіку автоматизованого проектування цифрових пристроїв на основі кінцевих автоматів [50, 51].

До основних етапів проектування відносяться наступні.

1. Концепція, тобто початковий етап, на якому оцінюється перспективність ідеї проекту з комерційної точки зору, розглядаються потреби ринку і можливості реалізації.

2. Специфікація, тобто розробка детальних вимог до функціональності, продуктивності і інших характеристик проекту, у тому числі створення високорівневого програмного прототипу пристрою, що проектується, який має назву моделі системного рівня, в якій можуть використовувати популярні мови системного рівня, такі як SystemC [52] і SystemVerilog [53].

3. Проектування RTL-моделі. На основі специфікації розробляється детальна архітектура системи що проектується: визначається структура і інтерфейси основних функціональних блоків, Модель з системного рівня покроково перетворюється на модель RTL-рівня (із застосуванням HDL - VHDL [54] чи Verilog [55]). Одним із способів опису HDL-моделей КА є автоматний шаблон, тобто спеціальна структура опису HDL-моделі, у якій функції переходів і виходів виділені в окремі процеси (процес), а призначення нового стану здійснюється у спеціальному процесі, пов'язаному з синхронізацією [56].

4. Функціональна верифікація. Розроблена HDL-модель перевіряється на наявність помилок в реалізації функціональних вимог шляхом використання TestBench – системи верифікації HDL-моделей [48, 57].

5. Логічний синтез. За допомогою спеціальних програмних засобів (синтезаторів) робоча RTL-модель перетвориться в модель вентильного рівня. Виконується повторна функціональна верифікація системи, що проектується, після синтезу, при якій поведінка вентильної моделі порівнюється з поведінкою RTL-моделі до синтезу [58].

6. Моделювання затримок (часова верифікація). На цьому етапі оцінюються часові характеристики вентильної моделі, враховуються елементні і транспортні затримки, сигнальні перегони.

7. Розміщення. Обирається технологія фізичної реалізації системи, що проектується, здійснюється реалізація елементів вентиляльної структури на реальні комірки чіпа, виконується оптимізація розміщення [59]. На цьому ж етапі в пристрій вносяться елементи діагностичної інфраструктури [60].

8. Фізична верифікація і конструкторське проектування. Проводиться аналіз споживання енергії, аналіз температурних аномалій, паразитних ємностей і інших низькорівневих ефектів [61]. На фінальній стадії чіп поміщається в ізольований корпус і розміщується на друкованій платі разом з іншими елементами системи, що проектується.

Ще одним класом автоматних моделей, які використовуються в системах логічного управління, є часові автомати в системах реального часу Система управління реального часу – це система, в якій результуюча дія (діяльність) залежить не лише від логічних значень простих управляючих дій, але і від часу, протягом якого ці дії виконуються. При описі поведінки систем управління реального часу модель кінцевого автомата розширюється введенням часової змінної, і вводиться поняття часового автомата (timed automata, timed FSM) [62].

Будь-який локальний цифровий пристрій, що реалізовує алгоритм обробки інформації або управління, може бути реалізований двома способами: апаратним або програмно-апаратним. При апаратному способі реалізації заданий алгоритм описується на мові опису апаратури (HDL) і синтезується інструментальними засобами САПР ПЛІС або ASIC (application-specific integrated circuit або інтегральна схема спеціального призначення). Перевагою такого підходу є апаратна гнучкість (можливість реалізувати будь-який алгоритм) і досить велика швидкодія. Основу сучасних спеціалізованих платформ проектування ПЛІС складають сімейства FPGA (Field Programmable Gate Array) фірми Xilinx (Virtex®, і Spartan®), які забезпечують значне поліпшення таких характеристик цифрової системи на кристалі, як продуктивність, споживання енергії при достатньо невисокій вартості [63]. При програмно-апаратному способі реалізації алгоритм

описується на апаратно-орієнтованій мові програмування (наприклад, на мові СІ із спеціальними бібліотеками) з урахуванням апаратної архітектури, на якій реалізовуватиметься задана програма. Це, як правило, різні сімейства мікроконтролерів, в тому числі з операційними системами реального часу [64].

1.3 Методи діагностування автоматних моделей

Важливим завданням при побудові систем автоматизованого контролю є забезпечення надійності їх функціонування, що неможливо без використання автоматичних систем технічного діагностування. Метою діагностування є забезпечення раціональної експлуатації електрообладнання при заданих показниках надійності і скорочення витрат на технічне обслуговування і ремонт (ТОР). Ця мета досягається шляхом управління технічним станом електрообладнання в процесі експлуатації, що дозволяє проводити ТОР відповідно до даних діагностування.

Основне завдання технічного діагностування полягає в отриманні достовірної інформації про технічний стан електро та газообладнання в процесі експлуатації. Вона вирішується на основі вимірювання, контролю, аналізу і обробки кількісних і якісних значень параметрів обладнання, а також шляхом управління обладнанням відповідно до алгоритму діагностування [65].

З часів класичної роботи Гілла [26], де визначені основні поняття теорії кінцевих автоматів та їх властивостей, діагностування кінцевих автоматів проводиться за процедурами, які отримали назву діагностичних експериментів (ДЕ) над автоматами.

Засадничою роботою, в теорії експериментів з автоматами, є робота Хенни [66]. У цій роботі запропоновано використовувати ряд характерних послідовностей, що дозволяють ідентифікувати таблицю переходів-виходів (ТПВ) справного автомата. Процедури, запропоновані в цій роботі, дають

найкращі результати для класу мінімальних сильнозв'язаних автоматів, що мають відмінні послідовності. У цій же роботі визначений клас встановлюючи, синхронізуючих і характеристичних послідовностей, що дозволяють будувати діагностичні експерименти для автоматів, що не мають характеристичних послідовностей. Надалі цей напрям отримав розвиток в роботах відомих вчених, таких, як Пархоменко П. П., Сагомонян Е.С., Тоценко В.Г., Сперанский Д.В., М. Abramović, М.А. Breuer [67-70]. Основна перевага цього напрямку полягає в простоті побудови перевіряючої послідовності по заданій ТПВ автомата і широкий клас несправностей, що виявляються, який обмежений лише несправностями, які збільшують число станів автомата.

Для побудови ДЕ над графовою моделлю керуючого автомата реалізується стратегія обходу усіх дуг графа переходів кінцевого автомата починаючи з початкової вершини за умови допустимості наявності більш, ніж однієї дуги (змішана стратегія). При цьому перевіряються усі поодинокі несправності переходів, а також справність функцій переходів КА [71, 72].

В [73] запропонована процедура автоматизації діагностування HDL-моделей кінцевих автоматів з використанням програми ASFTEST. Розглядається варіант відновлення графу переходів по HDL-моделі у формі автоматного шаблону та аналіз обходу усіх дуг графу для пошуку помилок проектування. В [74] розглянуті методи пошуку помилок проектування в HDL-моделях мікропрограмних автоматів. ДЕ проводиться шляхом обходу усіх дуг графа переходів КА, починаючи від початкової вершини, шляхом емуляції функцій операційного автомата в системі верифікації HDL-моделей (TestBench) в середовищі автоматизованого проектування Active-HDL. Зазначимо, що ця стратегія є дієвою тільки для «виключного» класу автоматів, тобто таких, у яких функції виходів для кожного стану є унікальними.

У [75] запропонований метод виявлення і локалізації помилок проектування в HDL-моделях кінцевих автоматів з довільними функціями

виходів. ДЕ проводиться шляхом обходу усіх дуг автомата Милі, починаючи з початкової вершини, у тому числі для автоматів «невиключного» класу. Для забезпечення повернення автомата з можливою помилкою проектування в початковий стан запропоновано використовувати синхронізуючі послідовності. ДЕ проводилися в системі верифікації VHDL-моделей (TestBench) у середовищі проектування Active-HDL.

При верифікації проектів критичних систем на ПЛІС застосовується метод засіву дефектів, тобто внесення в проект дефектів та аналіз реакцій на цей дефект. Це дозволяє прогнозувати виникнення аварійних ситуацій в критичних системах та розробляти засоби їх усунення. В [76] проведено аналіз існуючої техніки і інструментально-технічних засобів засіву дефектів вбудованих систем на ПЛІС. Запропоновано інструментальне засіб засіву і моделювання одиночних і множинних константних несправностей проектів на ПЛІС. Запропоноване засіб має низку переваг у порівнянні з існуючими програмними комплексами: не вимагає знань мов опису апаратури з боку користувача, універсальний формат представлення вхідних даних (EDIF 2.0.0), відкритий програмний код, можливість формування файлу "прошивки" для ПЛІС з інжектіваними дефектами.

В сучасних методах діагностування керуючих автоматів широко використовуються розширені автоматні моделі. Для тестування систем реального часу використовуються моделі Timed Finite State Machines [77]. Для опису часових аспектів в автоматній моделі використовуються, як правило, три параметри: часові обмеження (timing constraints), вхідні таймаути (timeouts) і вихідні затримки (output delays), іноді звані вихідними таймаутами. Вхідний таймаут визначає максимальний час очікування вхідної дії (події) для кожного стану автомата. Якщо вхідний сигнал не був поданий до закінчення таймауту, то автомат починає опитування вхідних змінних та може перейти в інший стан. Часові обмеження є інтервалами на переходах, що обмежують час, протягом якого перехід може бути виконаний. Вихідні затримки (вихідні таймаути, запізнення) відображають час, що витрачається

автоматом на виконання переходу, тобто вихідний сигнал з'являється на виході через інтервал часу, який визначається вихідною затримкою.

В [78] представлений метод отримання повного набору тестів для часового детермінованого FSM з тайм-аутами, з оцінкою верхньої межі числа станів і часових параметрів реалізації системи, що тестується. Показано, що тестовий набір, отриманий для відповідного класичного FSM набагато довше, ніж отриманий безпосередньо для FSM з тайм-аутами.

В [79] представлений метод отримання тестів відповідності з гарантованим покриттям відмов на основі моделі синхронізованого часового FSM (Timed FSM) з одним тактовим сигналом. Отримання тесту засноване на заданій області несправностей, яка дозволяє отримувати набори тестів з розумною довжиною. Область несправностей включає в себе кожну можливу помилкову реалізацію TFSM з відомими найбільшими межами часових обмежень і мінімальною тривалістю часових обмежень.

Відомо, що розширений кінцевий автомат (Extended FSM) є потужним формалізмом для моделювання цифрових пристроїв в системах логічного управління. На відміну від звичайних кінцевих автоматів, EFSM моделює окремі канали передачі даних і управління, що дозволяє представляти системи компактніше і, в деякому розумінні, знижує ризик конфлікту станів під час перевірки. Генерація тестів на основі моделей EFSM поширена при функціональній перевірці конструкцій апаратних засобів систем логічного управління [80]. Проте, проблема обходу графа станів EFSM здається нетривіальною із-за захисних умов, які дозволяють переходи між станами моделі. Запропонований алгоритм побудови шляху обходу графу станів, який дозволяє визначити перехід; який виконується шляхом аналізу залежностей управління і даних. Експерименти показали, що запропонований підхід забезпечує краще покриття переходів з коротшими тестовими послідовностями в порівнянні з відомими методами і забезпечує високий рівень покриття коду з точки зору вершин і дуг графа.

Окремим класом алгоритмів діагностування є діагностування

автоматних моделей у складі систем на кристалі (System-on-Chip, SoC). У [81] запропонована модульна конструкція SoC, яка відслідковує активації структур об'єктів інтелектуальної власності (IP-Cores) у складі SoC у тестових, налагоджувальних та функціональних режимах. Вдосконалена інфраструктура SoC Design-for-Test (DfT) із заходами безпеки, щоб перешкодити зовнішнім атакам на IP-Core у режимах тестування. Спочатку захищаються активи IP-Cores і SoC від атак у режимах тестування та налагодження, а потім повторно використовується інфраструктура DfT для виявлення атак у функціональному режимі. В [82] пропонується комплексний огляд різних аспектів безпеки SoC в режимі тестування, включаючи відомі моделі загроз, класифікацію атакуючих і існуючих методів, а також представлена концепція для побудови безпечної інфраструктури тестування SoC з акцентом на тестування вбудованих ядер.

Зі зростанням технологічної складності сучасні конструкції SoC продовжують збільшуватися в розмірах і залучають все більше IP-Cores. Таким чином, набагато складніше завершити тестування великих SoC в межах заданих часових витрат і вартості. Зазвичай автоматизований ієрархічний тест допомагає ефективно вирішити цю проблему, але для таких систем підготовка вхідних даних, особливо інформації на рівні IP-Cores та опису тестових послідовностей, зазвичай займає дуже багато часу. У роботі [83] представлено ефективне рішення для підготовки вихідних даних для ієрархічної системи тестування IP-Cores у складі SoC.

В [84] запропонований новий підхід к самодіагностиці «розумних» (smart) вбудованих систем управління та регулювання в різних галузях. Smart вбудовані системи функціонують в режимі жорстких обмежень на використання програмного та апаратного забезпечення. Введено поняття "self-x", як самоадаптивної розумної системи, що самодіагностується. Показано використання Кіберфізичеих Систем-на-кристалі (CPSoC) для smart-самодіагностики вбудованих систем.

1.4 Методи тестопридатного проектування цифрових пристроїв

Методи підвищення тестопридатності ЦП шляхом внесення до схемної реалізації апаратурної надлишковості досить розвинені і повсюдно використовуються при проектуванні.

У класичній роботі по розробці і класифікації структурних методів тестопридатного проектування цифрових систем [85] запропоновані основні визначення та процедури забезпечення тестопридатності. Схема є тестопридатною, якщо процедури генерації множини тестових наборів, оцінки їх ефективності і реалізації тестового діагностування можуть бути виконані за умови дотримання у встановлених межах фінансових витрат, витрат часу і значень показників, що характеризують пристосованість схеми до виявлення несправностей, пошуку місця несправностей і реалізації тестового діагностування. Тестопридатність забезпечується шляхом внесення до схеми пристрою, що проектується, апаратурної надлишковості, що забезпечує проведення діагностичних експериментів в межах витрат на проектування. Основний принцип методів структурного тестопридатного проектування схем полягає в організації сканування частини ЦП, що запам'ятовує, шляхом організації зсувних регістрів і побудову на їх основі сканованого шляху або просто «сканування». Метод сканованого шляху дозволяє вирішувати проблему тестопридатності шляхом зменшення складності структури схеми.

Практичні рекомендації по використанню структурних методів підвищення тестопридатності ЦП і побудови алгоритмів діагностування на їх основі детально викладені в роботах [13, 86]. Розглянуті різні підходи до проектування дискретних пристроїв, які враховують необхідність організації тестового діагностування пристрою в процесі виробництва і експлуатації. Приведено кількісне порівняння різних методів з урахуванням необхідної схемної надлишковості і отриманого об'єму. Наведені кількісні оцінки апаратної надлишковості для різних схемних структур. Викладені питання

оптимізації алгоритмів діагностування, принципи організації функціонального діагностування технічного стану складних безперервних і дискретних об'єктів, методи синтезу схем вбудованого контролю для дискретних пристроїв, питання організації самодіагностування і самовідновлення дискретних об'єктів, а також принципи побудови зовнішніх засобів тестового діагностування.

У класичній роботі [87] розглянуті методи вбудованого самотестування, як основа забезпечення тестопридатності цифрових пристроїв у процесі проектування. Показано застосування технології BIST (built - in self - test) при проектуванні цифрових систем на ПЛІС. Термін BIST може бути визначений, як здатність виробу (кристала, багатокристалльної збірки або системи) діагностувати свій власний стан, причому генерація вхідних сигналів і оцінка вихідних реакцій виконується самим пристроєм і не вимагається наявності зовнішньої апаратури. Вбудоване самотестування є підходом до діагностування цифрових пристроїв, пов'язаний з додаванням в схему додаткового тестового устаткування. У загальному випадку тестова структура, що вбудовується в пристрій, містить генератор тестів і аналізатор вихідної послідовності. Використання засобів BIST дозволяє спростити рішення багатьох завдань тестування: не вимагається генерація тестів і моделювання несправностей; тестування виконується на реальних частотах роботи пристрою, що перевіряється; значно знижується потреба у зовнішньому діагностичному устаткуванні; знижується час тестування за рахунок можливості паралельної перевірки окремих частин пристрою; спрощується перевірка на усіх структурних рівнях, від кристала до системи. При цьому витрати на додаткове тестове устаткування можуть досягати 30%. Розглянутий метод самотестування BILBO (Built-In Logic Block Observer), який заснований на використанні багаторежимного зсувного регістра з використанням лінійних зворотних зв'язків.

У роботах [88, 89] отримали подальший розвиток ідеї тестопридатного проектування на основі технології граничного сканування (Boundary Scan) з

використанням порту JTAG. Розглянуті стандарти граничного сканування IEEE 1149.x, IEEE 1500 і IEEE 1532, а також стандарт тестопридатного проектування IEEE 1687. В основу принципу Boundary Scan була покладена концепція розміщення послідовного зсувного регістру по межах пристрою. Таким чином, в пристрій додається 4 зовнішніх входи: TDI (Test Data In - вхід тестових даних), TDO (Test Data Out - вихід тестових даних), TMS (Test Mode Select - вибір тестового режиму) і TCK (Test Clock - тестова синхронізація). Декілька пристроїв на платі можуть бути об'єднані в єдиний шлях сканування. Це можна зробити шляхом з'єднання TDO з TDI наступного пристрою. Таким чином, на платі з'являється також 4 додаткові виводи: TDI і TDO, і загальні для усіх пристроїв TCK і TMS. Таким чином, технологія граничного сканування при відносно невеликих апаратурних витратах дозволяє забезпечити не лише 100% спостережуваність, але і 100% керованість входів/виходів пристроїв.

Функціональні методи тестопридатного проектування ЦП детально розглянуті в [68]. Для цифрових автоматів, представлених у формі ТПВ, введені поняття класів автоматів що діагностуються і запропоновані способи приведення ТПВ автоматів до вказаних класів. Розглянуті процедури проведення ДЕ з автоматами з використанням встановлюючих, синхронізуючих, діагностичних і розпізнаючих послідовностей. При функціональному підході до проектування (аналізу) цифрових пристроїв, заданих моделлю кінцевого автомата $W = \langle X, A, Y, \delta, \lambda \rangle$ підвищення тестопридатності автомата можливе тільки за рахунок введення апаратурної надлишковості шляхом розширення вхідного алфавіту X , алфавіту станів A або вихідного алфавіту Y . З одного боку це призводить до додаткових апаратурних витрат, але з іншого боку забезпечує збереження алгоритму функціонування автомата, заданого, як правило, його ТПВ або його графом переходів.

Питання проведення ДЕ з автоматами шляхом обходу усіх вершин і дуг графа переходів розглянуті в [69]. Дані оцінки довжини і повноти ДЕ при

різних варіантах обходу графа. Запропоновані процедури підвищення тестопридатності автоматів шляхом еквівалентних перетворень і розширення вхідного алфавіту.

У [85] введені поняття керованості та спостережуваності як основи оцінки структурної тестопридатності. Викладені принципи організації зсувних реєстрів в частині ЦП, що запам'ятовує, та побудови на їх основі сканованого шляху. Крім того, запропоновані модифікації відомих структурних методів побудови тестів при використанні методів сканування. В [90] тестопридатність цифрових схем на схемотехнічному рівні аналізується шляхом розрахунку діагностуємості на етапі побудови тестів методом активізації шляхів.

В технічних системах управління під досяжністю розуміють можливість (складність за часом та діями) встановлення системи в зазначений технічний стан [91]. У [92] поняття досяжності використовується для аналізу складності змістовних графових моделей кінцевих автоматів та визначення місць розміщення контрольних точок з метою встановлення асерційних конструкцій при проведенні верифікації. В [93] розглянуті способи розрахунку керованості та спостережуваності при аналізі графової моделі дорожнього руху для оптимізації маршрутів транспортних потоків. В [94] розглянуті питання використання кубічного числення і квантового комп'ютинга при тестопридатному проектуванні та діагностуванні цифрових Hardware-Software System.

В останні роки проводиться дуже багато досліджень в галузі тестопридатного проектування та вбудованого самотестування пристроїв, побудованих на великих інтегральних схемах.

В [95] показано, що архітектура вбудованого самотестування (BIST) використовується для ефективного тестування схем порівняно з тестуванням на основі сканування. У архітектуру BIST додаються мікросхеми для генерування тестових векторів та аналізу вихідних послідовностей. BIST зазвичай виконується з використанням генераторів псевдовипадкових

шаблонів (PRPGs).

В [96] констатується, що тестопридатне проектування є основним напрямком в промисловому виробництві SoC. Показано, що технологія DFT дозволяє суттєво знизити витрати на тестування при проектуванні SoC у асинхронному стилі. Логіка Sleep Convention Logic (SCL) - це новий багатообіцяючий стиль асинхронної логіки, заснований на більш відомому стилі NULL Convention Logic (NCL). В роботі проаналізовані різні несправності в конвеєрах SCL і запропонована методологія DfT на основі сканування, щоб зробити стиль SCL легкотестованим.

В [97] представлена методологія проектування цифрових схем на основі реверсивних вентилів, які б зберігали парність. Методологія забезпечує повне охоплення різних моделей несправностей шляхом виявлення однобітових несправностей.

У циклі проектування VLSI (Very-large-scale integration) широко використовуються методи введення додаткових контрольних точок для зменшення кількості тестових наборів на рівні регістрових передач. В [98] пропонується метод тестопридатного проектування, який дозволяє скоротити час на пошук місця розташування контрольних точок на рівні регістрових передач, що дозволяє одночасно тестувати якомога більше операційних блоків в режимі сканування.

В роботі [99] запропонована нова тестова архітектура, яка поєднує в собі переваги як методів сканування, так і вбудованого самотестування (BIST). Основна ідея полягає в тому, щоб записати (зберегти) всі необхідні стислі тестові послідовності в новій структурі ланцюжка сканування, і збережені дані можна було витягувати, відновлювати і розпаковувати в необхідні детерміновані послідовності з використанням вбудованого в чіп контролера тестування з декомпресором тестових послідовностей.

Робота [100] описує сучасний стан методів самотестування і показує, як ці методи повинні бути узагальнені для полегшення самодіагностики. Обладнання для самотестування має давню традицію як доповнення до

виробничих випробувань на основі тестових стимулів та аналізу відповідей. Сьогодні – це зріле поле і багато складних SoC мають вбудовані структури самотестування. Для SoC, оснащених самодіагностикою це ключова технологія, що дозволяє системі розрізнати правильну та помилкову поведінку.

В [101] розглянуті проблеми тестування біо-чипів для біохімічного аналізу. Дефектні чіпи ведуть до повторення експериментів, що небажано через високу вартість реагенту і обмежену наявність зразків. Проаналізовано вплив виробничих і фізичних дефектів на всі рівні архітектури біо-чипів і запропоновано використання теорії графів для максимізації покриття несправностей за допомогою використання додаткових контрольних тестових точок. Це є вагомим внеском у розробку процедур тестування систем біо-чипів.

В [102] для перевірки коректності роботи запропонована реалізація UART з вбудованими можливостями BIST з використанням технології FPGA. Проект виконаний на мові VHDL і функціонально перевірений шляхом моделювання коду в ModelSim від Mentor Graphics. Синтез ПЛІС був виконаний з використанням САПР Xilinx ISE.

При проектуванні сканованого шляху в сучасних дуже високошвидкісних схемах з мінімально можливою комбінаційної глибиною виникає погіршення продуктивності, яке викликане скануючим мультиплексором. В [103] запропонований новий підхід до організації сканування тригера, який усуває зниження продуктивності при послідовному скануванні. Пропонована конструкція видаляє скануючий мультиплексор з шляху функціонального сканування і підвищує швидкодію послідовного сканування.

Ще одним широко використовуваним в технічній діагностиці методом є сигнатурний аналіз. В роботі [104] продемонстровано реалізація економічних вбудованих функцій самоперевірки при апаратній реалізації криптоалгоритму AES в захищеній системі. Для самоперевірки пропонується

використовувати набір статистичних тестів, імітацію помилок і сигнатурний аналіз як спосіб аналізу еталонів.

При створенні критичних систем логічного управління в різних галузях техніки та транспорту широко застосовуються методи тестопридатного проектування. В [105] проаналізовано принципи і методи он-лайн тестування цифрових компонентів та систем. Охарактеризовано особливості діагностики систем, критично важливих для безпеки атомних електростанцій (АЕС). Запропоновано декілька методів і схем on-line тестування цифрових компонентів систем критичної безпеки. Ці методи засновані на використанні надлишковості природної інформації та зменшенні перевірки компонентів.

В [106] розглянуто вплив особливостей критично важливих контрольно-вимірювальних приладів та систем управління на перевірку їх цифрових компонентів. Проаналізовано природні ресурси, які можуть бути використані для підвищення надійності цифрових компонентів. Розглянуто природні ресурси цифрових компонентів, розроблених у вигляді багатоваріантної системи з сильно зв'язаними версіями.

В [107] на основі ресурсного підходу розглядаються перспективи розвитку компонентів комп'ютерних систем та їхнього робочого діагностування шляхом підвищення рівня паралелізму та наближеності. Запропоновано метод скануючого контролю для багатопоточної системи порозрядних конвеєрів, що виконують обробку наближених даних.

В [108] розглянуто особливості робочого діагностування цифрових компонентів у системах критичного застосування, що забезпечують функціональну безпеку об'єктів підвищеного ризику, включаючи кріогенну техніку. Показано доцільність розвитку робочого діагностування для порозрядних конвеєрних вузлів цифрових компонентів. В рамках моделі наближених даних отримали подальший розвиток методи контролю за нерівностями, що забезпечують робоче діагностування порозрядних конвеєрів. Запропоновано моделі результату, що служать йому границями, в межах яких результат визначається достовірним. Для трьох типів

порозрядних конвеєрних помножувачів визначені моделі доступу до даних контролю.

В [109] розглянута проблема прихованих несправностей, які можуть накопичуватися в цифрових компонентах систем безпеки при тривалому нормальному режимі та зменшувати відмовостійкість ланцюгів і безпеку системи в найбільш відповідальному аварійному режимі. Проблема прихованих дефектів вирішується шляхом реалізації цифрових компонент на основі побітових конвеєрів, реалізованих на ПЛІС за допомогою САD Altera. Несправності загальних сигналів, такі як порушення синхронізації, також можуть показувати приховану поведінку, яка не завжди виявляється при обробці послідовних кодів.

Робота [110] присвячена проблемі прихованих відмов, пов'язаних з розвитком комп'ютерних систем та їх перетворенням у безпечні контрольно-вимірювальні системи. Приховані несправності не дозволяють забезпечити функціональну безпеку шляхом побудови відмовостійкого цифрового компонента без урахування можливості їх перевірки. Підвищення діагностичних можливостей шляхом імітації аварії може спровокувати неприпустимий ризик переходу в аварійний режим. Запропоновані підходи до вирішення проблеми безпеки під час роботи з використанням методів поліпшення перевірки в нормальному режимі і з аналізом як нормального, так і аварійного режимів.

1.5 Постановка мети і завдань дослідження

Виходячи з огляду публікацій по розроблюваній тематики, можна зробити висновок про те, що питання автоматизованого тестопридатного проектування систем логічного управління на основі кінцевих автоматів є недостатньо вивченим і дослідження в даній області становлять наукову новизну. А саме: немає ефективних алгоритмів синтезу тестопридатних структур керуючих автоматів на основі HDL-моделей та використання

інструментальних засобів САПР РЕА.

Показано, що основним способом підвищення тестопридатності кінцевих автоматів є внесення апаратної надлишковості у схемну реалізацію цифрового пристрою, яка не має перевищувати режиму дублювання. Але ефективних методів автоматизації цього процесу а циклі автоматизованого проектування цифрових систем на ПЛІС не існує. Критичні системи логічного управління накладають свої особливості на процес проектування автоматних систем логічного управління.

Таким чином, актуальним і важливим науко-технічним завданням є розробка моделей і методів автоматизованого проектування та діагностування тестопридатних систем логічного управління критичного застосування на основі кінцевих автоматів.

Сутність дослідження полягає в підвищенні надійності функціонування критичних систем логічного управління за рахунок скорочення часу та підвищення якості методів діагностування у рамках проведення регламентних робіт щодо технічного обслуговування апаратної частини локальних систем управління технологічно відокремлених об'єктів електроенергетики та газопостачання, побудованих на основі кінцевих цифрових автоматів. Скорочення часу та підвищення якості методів діагностування досягається за рахунок розробки методів проведення тестопридатного проектування та діагностування автоматних систем логічного управління, побудованих на технологічній платформі ПЛІС. Тестопридатне проектування здійснюється з використанням інструментальних засобів САПР на основі мов опису апаратури.

Ринкова привабливість дослідження полягає у впровадженні моделей та методів тестопридатного проектування критичних систем логічного управління на основі керуючих автоматів приведе до зменшення часу на проектування та технічне обслуговування критичних автоматних систем логічного управління та кількості обслуговуючого персоналу.

Об'єкт дослідження – процес забезпечення тестопридатності цифрових

систем на основі кінцевих автоматів.

Предмет дослідження – моделі та методи тестопридатного проектування і діагностування автоматних систем логічного управління на ПЛІС.

Методи дослідження – апарати булевої алгебри, теорії автоматів і графів (подання моделей цифрових автоматів), технічної діагностики (підготовка та проведення діагностичних експериментів), засоби автоматизованого тестопридатного проектування цифрових автоматів (побудова, моделювання та синтез автоматних HDL-моделей).

Мета дослідження – розробка моделей і методів автоматизованого проектування та діагностування автоматних систем логічного управління на ПЛІС з використанням мов опису апаратури для скорочення часу технічного обслуговування критичних систем.

Задачі дослідження:

- 1) визначити особливості діагностування критичних апаратних систем логічного управління;
- 2) удосконалити методи неруйнівних діагностичних експериментів для моделей кінцевих автоматів;
- 3) розробити методи побудови апаратних систем підтримки проведення діагностичних експериментів в автоматних системах керування;
- 4) удосконалити методи оцінки тестопридатності графових моделей кінцевих автоматів;
- 5) розробити методи підвищення тестопридатності моделей кінцевих автоматів за рахунок розширення вхідного алфавіту;
- 6) реалізувати розроблені моделі в програмних модулях побудови тестопридатних автоматів в САПР ПЛІС.

РОЗДІЛ 2

МОДЕЛІ ТА МЕТОДИ ДІАГНОСТУВАННЯ ЦИФРОВИХ КЕРУЮЧИХ АВТОМАТІВ

Другий розділ присвячено розробці моделей та методів проведення діагностичних експериментів стосовно керуючих автоматів, наданих мовами опису апаратури, з використанням вбудованих пристроїв діагностування.

В ході проведення досліджень мають бути вирішені такі завдання :

1) визначити особливості діагностування критичних апаратних систем логічного управління, створити моделі пристроїв керування мовами опису апаратури для локальної системи управління газорегуляторними пунктами;

2) удосконалити методи проведення неруйнівних діагностичних експериментів для моделей кінцевих автоматів з використанням синхронізуючих послідовностей, які забезпечують повернення керуючого автомату в початковий стан;

3) розробити методи проектування вбудованих апаратних систем діагностування для підтримки проведення неруйнівних діагностичних експериментів в автоматних системах управління та забезпечити апаратну надлишковість, яка не перевищує режим дублювання.

Основні результати досліджень опубліковані в роботах [115, 116, 119, 126, 129].

2.1 Моделі автоматного пристрою керування ГРП

В даний час в електроенергетиці та газопостачанні відбуваються суттєві зміни, пов'язані з впровадженням нових технологій автоматизації та енергозбереженні. На рівні підприємств і територій використовуються автоматизовані системи управління технологічними процесами, які, в основному, управляють інформаційними потоками. У системах газопостачання використовуються автоматизовані системи оперативно-

диспетчерського управління, а в їх складі спеціальні локальні системи управління і регулювання, розташовані на віддаленій місцевості, наприклад на газорозподільних вузлах, газорегуляторних пунктах, пунктах обліку газоспоживання та енергоспоживання тощо.

Системи автоматичного регулювання (САР) призначені для підтримки постійної або змінної по заданому закону деякої керованої величини. Системи автоматичного управління (САУ) здійснюють сукупний вплив на об'єкт, обраний з множини можливих впливів, спрямованих на досягнення певного критерію керування. У загальному випадку, САР і САУ можуть будуватися як на основі локальних засобів автоматичного регулювання, так і з застосуванням цифрових САУ. Такі локальні системи, як правило, реалізуються на інтегральних схемах, виготовлених на замовлення, на ПЛІС або мікроконтролерах.

Важливим завданням при побудові САУ є забезпечення надійності їх функціонування, що неможливо без використання автоматичних систем технічного діагностування. Особливістю технічного діагностування систем газопостачання є те, що для проведення технічного діагностування зазначені системи не можна зупинити, а тим більше демонтувати. Найбільш важливим компонентом систем управління при проведенні діагностики є цифрові блоки логічного управління.

При проектуванні сучасних систем технічного діагностування широко використовуються комп'ютерні технології автоматизованого проектування з застосуванням мов опису апаратури і сучасної технологічної бази. Даний підхід дозволяє реалізувати систему діагностування віддаленого пункту управління будь-якої енергетичної системи без участі людини і без відключення основної системи управління на тривалий час.

Ефективність функціонування автоматизованих систем (АС) в значній мірі залежить від того, наскільки повно, при виборі методів і засобів контролю, були враховані особливості зазначених систем як об'єктів контролю. Без урахування цих особливостей АС практично неможливо

встановити для кожної з них роль і місце контролю в управлінні її функціонуванням. Специфіка АС як об'єктів контролю впливає на вибір методів і видів контролю, а також показників якості їх функціонування. Вона знаходить своє відображення в принципах формалізації і змістовного опису процесів контролю, в принципах синтезу функцій працездатності АС.

Метою діагностування є забезпечення раціональної експлуатації газорегуляторного обладнання та електрообладнання при заданих показниках надійності і скорочення витрат на технічне обслуговування і ремонт (ТОР). Ця мета досягається шляхом управління технічним станом обладнання в процесі експлуатації, що дозволяє проводити ТОР відповідно до даних, отриманих в процесі діагностування. Основне завдання технічного діагностування полягає в отриманні достовірної інформації про технічний стан електро та газообладнання в процесі експлуатації. Вона вирішується на основі вимірювання, контролю, аналізу і обробки кількісних і якісних значень параметрів обладнання, а також шляхом управління обладнанням відповідно до алгоритму діагностування [65].

Аналіз причин виникнення дефектів обладнання показує, що технічний стан кожного з них характеризується як тільки йому притаманними індивідуальними показниками, так і загальними ознаками. Для кожного виду обладнання характерні свої типові дефекти, які багаторазово зустрічаються в експлуатації. Об'єднавши всі дефекти і ознаки їх появи в окремі групи, отримаємо структуру діагностування обладнання, що складається з трьох рівнів і підсистем: перевірки функціонування, виявлення дефектів, оцінки і прогнозування працездатності. При цьому на кожному наступному рівні використовуються результати попередніх діагностичних експериментів.

До апаратурних засобів діагностування відносяться різні пристрої: прилади, пульти, стенди, спеціальні промислові комп'ютери. Апаратурні пристрої, які складають з об'єктом діагностування, конструктивно єдине ціле, є вбудованими апаратурними засобами діагностування. Прикладами подібних засобів можуть бути електровимірювальні прилади (струму,

напруги, потужності, частоти та інші), газовимірювальні прилади (тиску, температури, вологості газу), пристрої індикації технічного стану елементів (реле, світловипромінюючі діоди, неонові лампи і тощо), пристрої контролю ізоляції, витоків газу тощо.

Якщо в схемах експлуатації газообладнання непередбачено вбудовані засоби діагностування або їх виявляється недостатньо для діагностування з необхідною глибиною, то застосовують зовнішні апаратурні засоби діагностування, виконані окремо від конструкції обладнання і підключаються до нього лише в процесі діагностування. Найпростішими прикладами зовнішніх апаратурних засобів в електроенергетиці можуть бути комбіновані прилади для вимірювання в колах постійного і змінного струму, тестери логічного стану, електронно-променеві та цифрові осцилографи, переносні вимірювальні комплекти і тощо.

Зовнішні спеціалізовані засоби діагностування – це пристрої, що використовуються, наприклад, для перевірки працездатності окремих елементів або вузлів газо та електрообладнання на стадіях технічного контролю після виконання ремонтних робіт.

Основним способом реалізації систем логічного управління (де керуючі сигнали формуються в алфавіті $\{0, 1\}$) виходячи з порівняння вимірюваних або обчислюваних параметрів з межами їх допустимих значень, є автоматні системи управління на основі композиції керуючого і операційного автоматів. ОА в таких системах, як правило, перетворює аналогові електромеханічні сигнали датчиків в двійковий алфавіт, а КА реалізує власне алгоритм логічного управління [29].

В основі функціонування автоматних моделей систем логічного управління лежить поняття «кінцевий автомат». Кінцевим автоматом називається математична модель синхронної системи для перетворення вхідної інформації, представлені кінцевим алфавітом у вихідну інформацію, представлену також кінцевим алфавітом через кінцеве число стійких станів [26]. Абстрактний кінцевий автомат представляється шестіркою

$W = \langle X, A, Y, \delta, \lambda, a_0 \rangle$, де $X = \{x_1, x_2, \dots, x_m\}$ – множина букв вхідного алфавіту; $A = \{a_1, a_2, \dots, a_n\}$ – множина станів автомату; $Y = \{y_1, y_2, \dots, y_r\}$ – множина букв вихідного алфавіту; $\delta(a_i, x_k) = a_j$ – функція переходів автомату, $\lambda(a_i, x_k) = y_\alpha$ – функція виходів автомату, a_0 – початковий стан автомату.

Структурні моделі автоматів, які використовуються в задачах логічного управління, можуть розглядатися як пристрої керування. З точки зору реалізації алгоритму логічного управління структурний автомат це модель пристрою реального часу, яка характеризується видачею визначених вихідних сигналів в визначені моменти часу. Структурний автомат функціонує у автоматному часі, який вимірюється в тактах $\{t, t+1, t+2\}$, тобто автомат переходить з одного стану до іншого за один автоматний такт. За один автоматний такт автомат формує множину вихідних сигналів та обчислює значення наступного стану.

У аналітичному вигляді модель структурного автомату має вигляд $Y(t) = g(X(t), Z(t))$, $Z(t+1) = f(X(t), Z(t))$, де g – функція виходів структурного автомату, f – функція переходів структурного автомата. При цьому $Z(t+1) \equiv Z(t)$, але у наступному автоматному такті.

Таким чином, апаратна реалізація моделі структурного автомата складається з комбінаційного і послідовнісного компонентів. Послідовнісний компонент містить елементи пам'яті, такі як синхронні тригери, які запам'ятовують значення внутрішніх змінних (стан) і дозволяють змінювати його синхронно. Комбінаційний компонент складається з логічних елементів, які реалізують дві логічні функції: функцію виходів, яка обчислює значення вихідних сигналів, і функцію переходів, яка обчислює нові значення елементів пам'яті або внутрішніх змінних (тобто значення наступного стану) [111]. На рис. 2.1 наведена графічна інтерпретація апаратної реалізації структурного автомату.

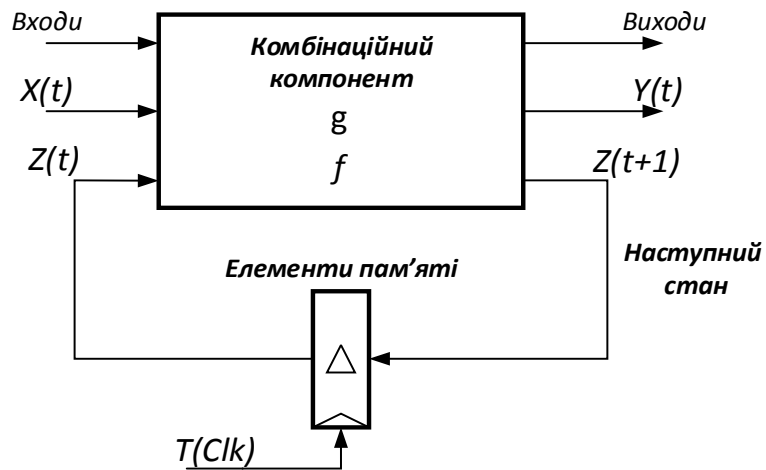


Рисунок.2.1 – Модель структурного автомату

В сучасному циклі проектування ЦП діагностичне забезпечення розробляється ще на етапі проектування, на що витрачається більше половини загального часу проектування. Основною формою опису проектів ЦП в САПР РЕА є мови опису апаратури, тому для розробки діагностичного забезпечення для реального ЦП, його треба розробляти для моделі ЦП, яка написана на мові опису апаратури, тобто для HDL-моделі.

Можливі помилки проектування в HDL-моделях визначаються стилем опису HDL-коду. Під помилкою проектування вважається визначення помилки в HDL-операторі, яка не відноситься до класу синтаксичних і порушує алгоритм функціонування моделі пристрою, заданий специфікацією. Виділення фрагментів HDL-коду, що описують поведінку кінцевих автоматів, стилем «автоматний шаблон», дозволяє визначити помилку проектування типу «неправильний перехід у графі переходів автомата», що відповідає помилці у виборі поточного стану в операторі when, помилці вибору наступного стану в функції переходів (a_i замість a_j), помилці в операторі if() при аналізі вхідного сигналу, помилці в призначенні вихідного сигналу. Для проведення діагностичного експерименту з пошуку помилок проектування реалізується стратегія обходу всіх дуг графа переходів кінцевого автомата, починаючи з початкової вершини. При цьому перевіряються всі поодинокі несправності переходів, а також справності

функцій автомата, що забезпечують ці переходи [72].

ДЕ над HDL-моделлю кінцевого автомата полягає в подачі на неї вхідних сигналів, відповідно до обраної стратегії обходу змістовного графа переходів, отриманні вихідних реакцій на Waveform і порівняння отриманих реакцій з еталоном. На підставі цього робиться висновок про відповідність HDL-моделі специфікації. ДЕ проводиться з використанням системи верифікації HDL-моделей (TestBench) в середовищі проектування, наприклад, Active-HDL. При проведенні ДЕ в простих HDL-моделях КА, подача вхідних впливів і порівняння отриманих реакцій з еталонами не представляє особливих труднощів, навіть в режимі візуального порівняння з Waveform, так як тестові дані подаються, безпосередньо, на входи автомата, а реакції знімаються з його виходів.

Сукупність об'єкта управління та технічних пристроїв, призначених для нього, називається системою автоматичного управління (регулювання) (САУ, САР). Основне завдання САР полягає в тому, щоб на основі інформації про об'єкт виробити керуючі сигнали, що дозволяють підтримувати об'єкт в стабільному стані або перевести його в новий стабільний стан. Технічні пристрої, що входять до САР, включають в себе: датчики; пристрої, що визначають закон функціонування об'єкта; регулятори, що виробляють керуючі сигнали по необхідному закону керування; керівні органи і виконавчі механізми.

Як об'єкт управління (ОУ) розглянемо газорегуляторний пункт (ГРП) у складі автоматизованої системи оперативно-диспетчерського управління об'єктами газопостачання, який розташовується, як правило, на віддаленій території і функціонує без присутності кваліфікованого персоналу. Газорегуляторним пунктом називається комплекс технологічного устаткування і пристроїв, призначений для пониження вхідного тиску газу до заданого рівня і підтримки його на виході постійним. Для забезпечення моніторингу територіально віддалених ГРП з локальними системами управління важливо впровадити комплекси самодіагностики та телеметрії .

Для цього у складі кожного ГРП є контролюючі пункти (КП), за допомогою яких буде забезпечений моніторинг і сигналізація територіально-віддалених ГРП з локальними системами управління. Спрощена функціональна схема роботи системи локального управління КП ГРП представлена на рис 2.2.

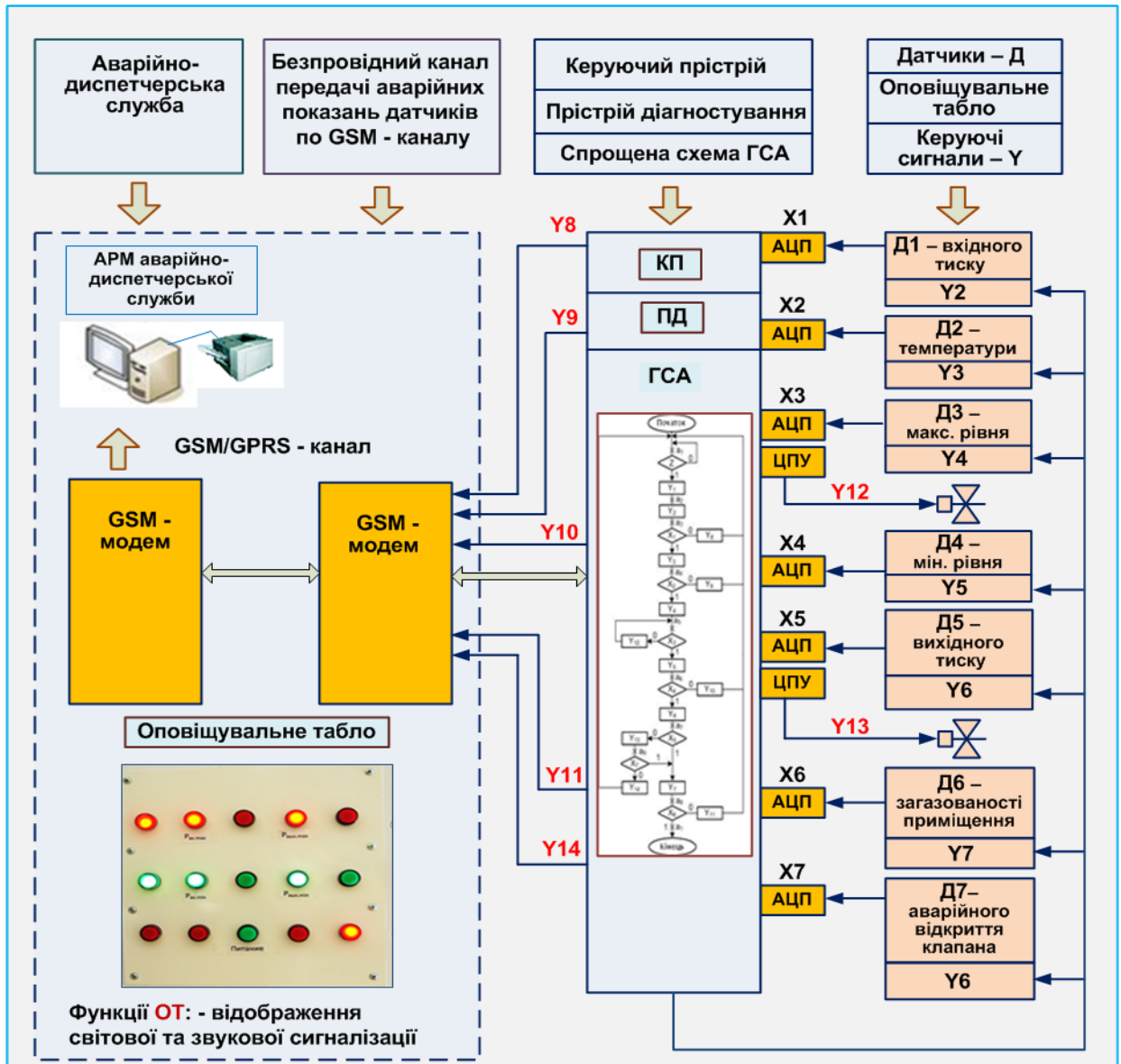


Рисунок 2.2 – Функціональна схема роботи системи локального управління КП ГРП

Технологічно КП ГРП представляють собою контролери, які зчитують інформацію про технологічний режим у вигляді сигналів, що надходять від аналогових датчиків тиску й температури, лічильників газу, дискретних датчиків та датчиків загазованості. На рівні контролерів реалізована обробка

даних, виявлення аварійних ситуацій і локальні алгоритми технологічного контролю й керування. Локальні системи управління КП ГРП (системи технологічної сигналізації) в реальному часі відстежують стан охоронних датчиків, систем контролю технологічних процесів на об'єкті, і в разі аварії, проводять оповіщення персоналу за допомогою тривожної сигналізації та оповіщення оператора аварійно-диспетчерської служби (АДС) через GSM-канал. Інформація про роботу ГРП передається по бездротовому каналу зв'язку та відображається на цифровому табло індикації. Табло індикації складається зі світлодіодів зеленого і червоного кольору, що відповідає робочому (зелений) і аварійному (червоний) режиму роботи обладнання ГРП [17].

З точки зору теорії управління КП ГРП являє собою розімкнуту систему логічного управління (регулювання) на основі керуючого автомата. Керуючі сигнали (Y_i) ініціюють зчитування аналогових сигналів відповідних датчиків, перетворення їх в логічні значення $\{0, 1\}$, які в свою чергу є сповіщувальними сигналами для керуючого автомата. Таким чином реалізується алгоритм регулювання КП ГРП.

Словесний опис спрощеного алгоритму функціонування КП ГРП і опис використовуваних датчиків представлено нижче:

1. Датчик вхідного тиску P_1 (X_1).

Тиск на вході контролюється за допомогою аналогового датчика тиску P_1 (зняття показань Y_2), який працює в діапазоні $0,3\text{МПа} \leq P_1 \leq 0,5\text{МПа}$ ($X_1=1$). Якщо тиск більше або менше зазначеного, то $X_1=0$ і на табло індикації буде горіти світлодіод червоного кольору (Y_8).

2. Датчик температури T_1 (X_2).

Датчик температури T_1 (зняття показань Y_3), працює в діапазоні $10^\circ\text{C} \leq T_1 \leq 25^\circ\text{C}$ ($X_2=1$). При збільшенні або зменшенні температури більше заданої ($X_2=0$) на табло індикації буде переданий аварійний сигнал (Y_9).

3. Датчик контролю максимального рівня газоконденсату H_1 (X_3).

Невід'ємною частиною обладнання ГРП є вузол очищення газу від

рідких домішок і механічних частинок. Вузол очистки складається з двох сепараторів, які підключені до системи газопроводів паралельно, що дає їм можливість працювати незалежно один від одного. У сепараторі відбувається відділення газу від рідкої фракції. Очищений газ проходить через верхню частину сепаратора в газопровід і далі надходить на регулятор тиску. Контроль максимально можливого заповнення сепаратора здійснюється за допомогою датчика H_1 (зняття показань Y_4), Датчик H_1 працює в діапазоні $40\% \leq H_1 \leq 60\%$ ($X_3=1$).

При досягненні рідкою фракцією рівня 60% – датчик сигналізує про переповнення ($X_3=0$) і на електромагнітний соленоїдний клапан скидання рідини надходить керуючий сигнал на його відкриття (Y_{12}). Рідина надходить в роздільну ємність для подальшої обробки та утилізації. Як тільки рівень рідини в сепараторі зменшиться до 40%, електромагнітний клапан закриється.

4. Датчик контролю мінімального рівня газоконденсату H_2 (X_4)

При досягненні рівня рідини 15% спрацює датчик H_2 (зняття показань Y_5) мінімального рівня ($X_4=0$) і на табло індикації буде переданий аварійний сигнал (Y_{10}). Зниження рівня рідини в сепараторі до 15% ($X_4=1$) може відбутися тільки в разі зниження тиску на вході, тобто $P_1 \leq 0,3$ МПа.

5. Датчик контролю вихідного тиску P_2 після регулятора (X_5)

Регулятор тиску на ГРП забезпечує зниження тиску газу до необхідного ($P_2=0,1$ МПа) і автоматично підтримує його в рамках цього тиску (зняття показань Y_6). Датчик вихідного тиску P_2 після регулятора працює в діапазоні $0,05 \text{ МПа} \leq P_2 \leq 0,125 \text{ МПа}$ ($X_5=1$). При підвищенні вихідного тиску $P_2 \leq 0,125 \text{ МПа}$ ($X_5=0$) спрацьовує запобіжний скидний клапан (Y_{13}) і ініціюється датчик передачі аварійного сигналу при підвищенні тиску (X_7).

6. Датчик загазованості приміщення АГРС CH_4 (X_6).

Загазованість приміщення ГРП може виникнути в разі витоків газу через фланцеві з'єднання або через сальникові ущільнення газового обладнання (зняття показань Y_7). У безаварійному режимі $X_6=1$.

Вибухонебезпечна концентрація газу в приміщенні знаходиться в межах від 5% до 15%. Датчик загазованості вже при концентрації 1% ($X6=0$ при $CH_4 \geq 1\%$) передає аварійний сигнал на табло індикації (Y11).

7. Датчик приводу аварійного відкриття клапана ЗСК при підвищенні тиску (X7). Датчик, контролює тільки підвищення тиску. При $P_2 \geq 0,125$ МПа, що відповідає підвищенню тиску на 25% від робочого ($X7 = 0$), на табло індикації буде горіти світлодіод червоного кольору і спрацює звукова сигналізація (Y14).

На рис. 2.3 представлена спрощена граф-схема алгоритму роботи керуючого автомата КП ГРП.

Для перетворення ГСА в автоматну модель, необхідно відзначити стани автомата. Як автоматну модель будемо розглядати автомат Мілі. Його станами на ГСА позначаються входи вершин, наступних за операторними вершинами [111]. Після позначення станів на ГСА будується змістовний граф переходів автомата Мілі. Вершинам графа переходів відповідають стани автомата a_i , дугам відповідають переходи між станами. На дугах автомата моделі Мілі вказуються функції умов переходів $f_j(x_1, x_2, \dots, x_n)$ та вихідні (керуючі) сигнали. На основі автоматної моделі Мілі будується граф переходів (State Diagram) керуючого автомата КП ГРП, який представлений на рис. 2.4.

Слід зазначити, що State Diagram є найбільш придатною формою представлення моделі КА для побудови програмного коду в стилі автоматного програмування. Крім того, стиль автоматного програмування на мовах опису апаратури орієнтований на опис моделей КА та враховує особливості синтезованої підмножини HDL-коду.

Для подальшого автоматизованого проектування та проектування діагностичного забезпечення граф переходів керуючого автомата КП ГРП представляється на мові опису апаратури VHDL у формі дво процесного автоматного шаблону. Фрагмент VHDL-моделі автомата Мілі, представлений в лістингу 2.1.

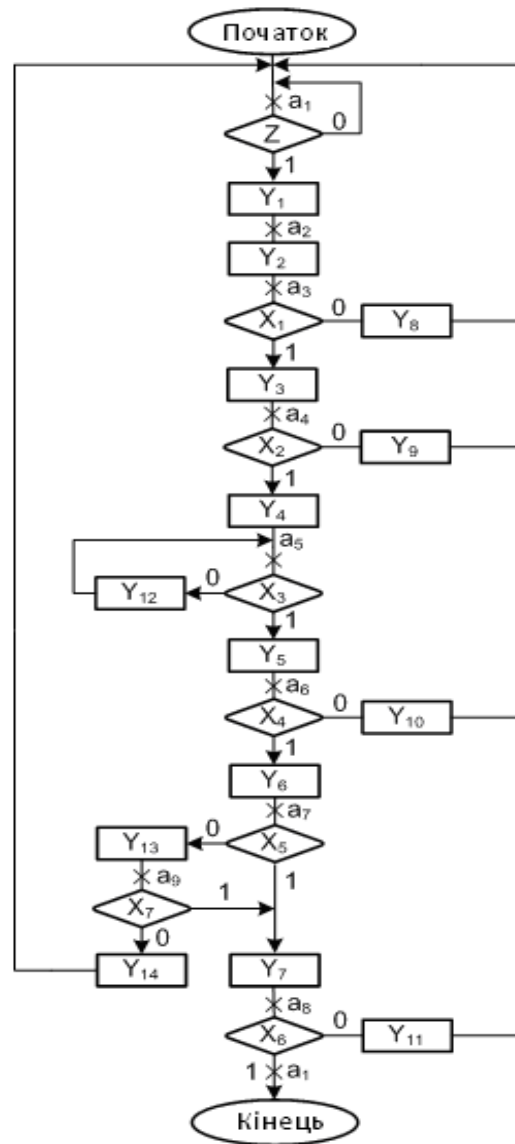


Рисунок 2.3 – Граф-схема алгоритму роботи КА КП ГРП

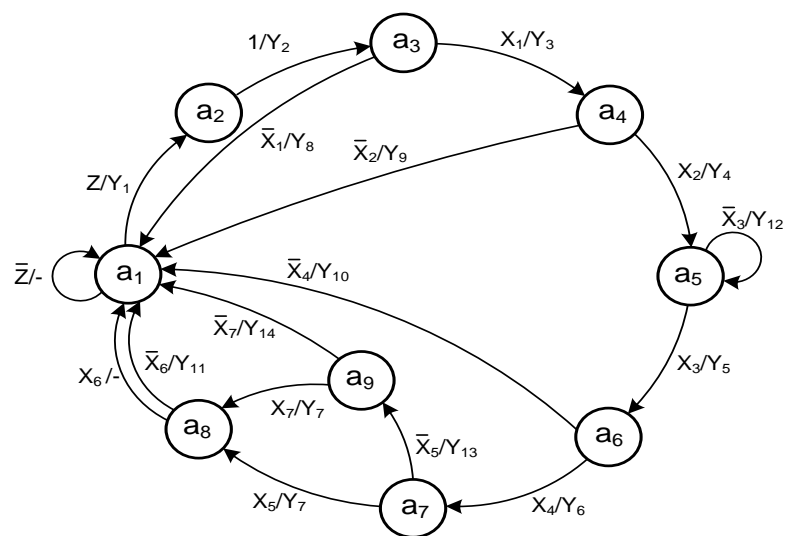


Рисунок 2.4 – Граф переходів керуючого автомату КП ГРП

Лістинг 2.1 – Фрагмент VHDL-модель керуючого автомата КП ГРП

```

begin
State_CurrentState: process (clk, reset)
begin
    if rising_edge(clk) then
        if reset='1' then State <= A1;
        else State <= NextState;
        end if;
    end if;
end process;

State_NextState: process (X, State)
begin
Y<= (others=>'0');
case State is
    when "0001" =>
        if (X(0)='1') then NextState <= A2;  Y(1) <= '1';
        elsif (X(0) = '1') then NextState <= A1;
        else NextState <= A1;
        end if;
..... . . .
    when "0110" =>
        if (X(4)='1') then NextState <= A7;  Y(6) <= '1';
        elsif (X(4) = '0') then NextState <= A1; Y(10) <= '1';
        else NextState <= A1;
        end if;

```

Повна VHDL –модель керуючого автомату КП ГРП та результати її верифікації наведені у додатку Г.

2.2 Діагностичні експерименти над керуючими автоматами

У відповідності до технічних вимог з побудови систем управління (регулювання) КП ГРП вони мають бути обладнані системами самодіагностики [18], тобто перевірки правильності реалізації алгоритму регулювання. Зазначені системи можуть функціонувати у двох режимах.

1. Віддалений режим, У цьому режимі при проведенні регламентних робіт з технічного обслуговування ГРП запускається режим діагностування керуючого автомату в системі управління ГРП. Процес діагностування керуючого автомату (діагностичний експеримент) пов'язаний з різними варіантами обходу його графу переходів, але без використання сигналів самих датчиків. Керуючі сигнали з датчиків можуть імітуватися спеціальним пристроєм діагностування, або може реалізуватися спеціальний діагностичний режим роботи пристрою керування без використання сигналів реальних датчиків. При проведенні діагностичного експерименту керуючий автомат має повернутися в початковий стан навіть при наявності несправностей у керуючому пристрої.

2. Локальний режим. При проведенні регламентних робіт з технічного обслуговування безпосередньо на ГРП у спеціальному режимі керуючий автомат може бути примусово встановлений в довільний стан (ініційована робота відповідного датчика) і спеціаліст з обслуговування контрольно-вимірювальних пристроїв може перевірити працездатність зазначеного приладу контролю в автономному режимі.

При цьому слід зазначити наступне. Виходячи з того, що режим діагностування проводиться незалежно від алгоритму функціонування пристрою керування, часові параметри функціонування керуючого автомату в цьому режимі не враховуються і моделі часових автоматів не розглядаються. Діагностування керуючого автомату у режимі реального часу може бути предметом подальших досліджень.

Діагностичний експеримент над HDL-моделлю керуючого автомата

полягає в подачі на неї вхідних впливів, відповідно до обраної стратегії обходу змістовного графа переходів, отриманні вихідних реакцій на Waveform і порівняння отриманих реакцій з еталоном. На підставі цього робиться висновок про відповідність HDL-моделі специфікації. ДЕ проводиться з використанням системи верифікації HDL-моделей (TestBench) в середовищі проектування Active-HDL. При проведенні ДЕ в простих HDL-моделях КА, подача вхідних впливів і порівняння отриманих реакцій з еталонами не представляє особливих труднощів, навіть в режимі візуального порівняння з Waveform, так як тестові дані подаються, безпосередньо, на входи автомата, а реакції знімаються з його виходів.

Для побудови тесту реалізується стратегія обходу всіх дуг графа переходів кінцевого автомата починаючи з початкової вершини за умови допустимості наявності більш, ніж однієї дуги $a_i \Rightarrow a_j$ (змішана стратегія). Даний підхід передбачає проведення так званого «неруйнівного» експерименту, в якому в кінці кожної перевірки автомат логічно або примусово повертається в початковий стан. При цьому перевіряються всі поодинокі несправності переходів, а також справності функцій автомата, що забезпечують ці переходи. ДЕ над HDL-моделлю кінцевого автомата полягає в подачі на неї вхідних впливів відповідно до обраної стратегії обходу змістовного графа переходів, отриманні вихідних реакцій у вигляді вихідних сигналів або списку станів автомата на Waveform або списку обходу графа в файлі і порівняння отриманих реакцій з еталоном візуальним або програмним шляхом. На підставі цього робиться висновок про відповідність HDL-моделі специфікації. ДЕ проводиться з використанням системи верифікації HDL-моделей (TestBench) в середовищі проектування Active-HDL, моделі автоматів представлені на мові VHDL.

На підставі стратегії обходу всіх дуг графа КА (рис. 2.4) будується алгоритм діагностування з гарантованою повнотою щодо одиночних несправностей переходів (наприклад, перехід $a1 - a2$ замість $a1 - a3$), представлений на рис.2.5 у вигляді бінарного дерева рішень [74].

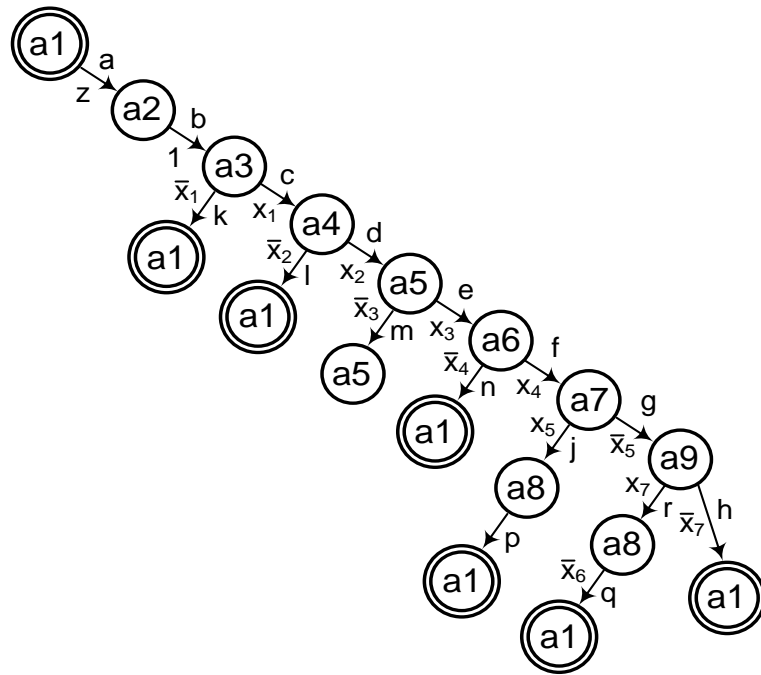


Рисунок 2.5 – Дерево рішень для графа переходів КА КП ГРП

По дереву рішень будуються варіанти обходу дуг графа. При цьому слід враховувати, що проводиться так званий «неруйнівний ДЕ», коли обхід дуг графа починається з початкової вершини і в ній же закінчується. Варіанти обходу графа переходів КА представлені на рис. 2.6.

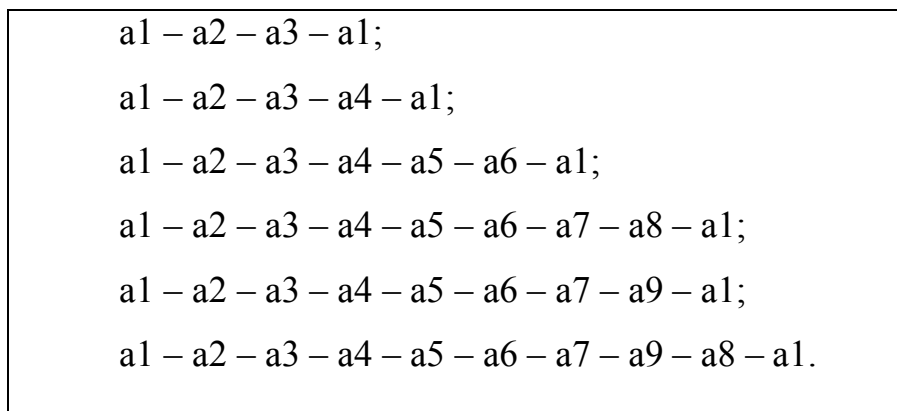


Рисунок 2.6 – Варіанти обходу графа переходів КА

На рис. 2.7 показаний результат моделювання режиму роботи КА КП ГРП, який описує послідовність переходів $a1-a2-a3-a4-a5-a6-a1$. Моделювання здійснювалося в середовищі проектування Active-HDL.

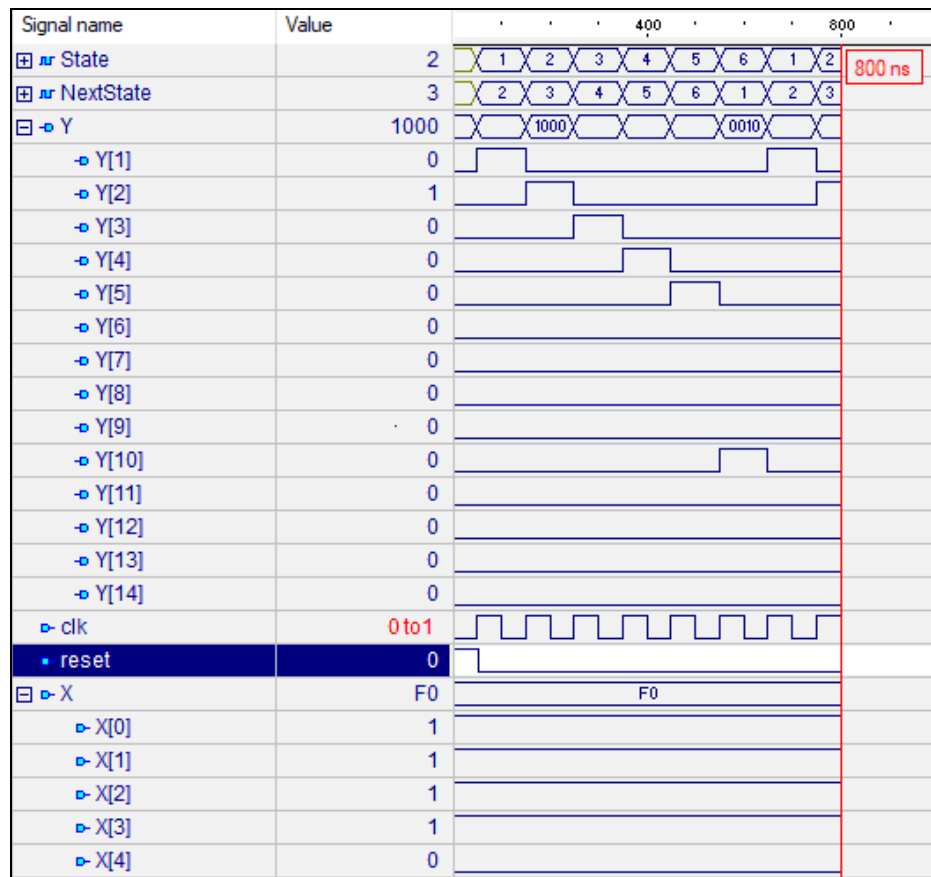


Рисунок 2.7 – Результати моделювання роботи КА КП ГРП

Перехід зі стану в стан відповідає за ініціацію блоків і датчиків, описаних в лістингу 2.1. Відбувається ініціація автомата (запуск різних блоків для початку роботи УА ($a1 - a2$; $X(0) = 1 \Rightarrow Y1 = 1$)) опитування датчика тиску ($a3 - a4$; $Y2 = 1$), температури ($a4 - a5$; $X(1) = 1 \Rightarrow Y3 = 1$) та газоконденсату ($a5 - a6$; $X(3) = 1 \Rightarrow Y4 = 1$). Після опитування кожного датчика відбувається порівняння їх показників з необхідними для подальшої роботи. Позитивна перевірка ($X(1) = X(2) = X(3) = 1$) показує, що показники не перевищують допустимі і автомат далі продовжує коректну роботу. Перевірка наступного датчика мінімального рівня газоконденсату не проходить ($a6 - a7$; $X(4) = 0$) і виникає аварійна ситуація ($Y(10) = 1$), яка повертає автомат в початковий стан ($a1$).

Апаратні витрати по синтезу моделі керуючого пристрою засобами САПР XILINX ISE в ПЛІС Spartan 3E наведені в таблиці 2.1.

Таблиця 2.1 – Результати синтезу КА КП ГРП

Selected Device : 3s500efg320-5	ПЛІС Spartan 3E
Number of Slices	10 out of 4656
Number of Slice Flip Flops	9 out of 9312
Number of 4 input LUTs:	18 out of 9312
Number of IOs	24
Number of bonded IOBs:	24 out of 232
Number of GCLKs	1

Основним завданням при проведенні «неруйнівного» діагностичного експерименту є можливість повернення автомата в початковий стан незалежно від результату чергової перевірки. Для цих цілей доцільно використовувати синхронізуючі послідовності кінцевих автоматів [112].

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

Якщо автомат $W = \langle X, A, Y, \delta, \lambda, a_0 \rangle$ заданий таблицею переходів-виходів або графом переходів, то з визначення випливає, що автомат має синхронізуючу послідовність x_s таку, що $\delta(a_i, X_j) = a_0, \forall a_i \in A, a_0 \in A$. Множина переходів $\delta(a_i, X_s) = a_0, \forall a_i \in A$, автомата визначає відображення множини його станів A в якийсь певний стан a_0 при подачі на автомат вхідної послідовності X_s , тобто $a_i \xrightarrow{X_s} a_0$.

Синхронізуюча послідовність для заданого автомата може бути знайдена з синхронізуючого дерева, яке є деревом наступників, побудованим за певними правилами [75]. На рис. 2.8 наведено фрагмент побудованих синхронізуючих послідовностей для КА КП ГРП за моделлю Мілі та забезпечуючих повернення до стану a_1 . Повне синхронізуюче дерево для цього стану та синхронізуючі послідовності наведені у додатку Д.

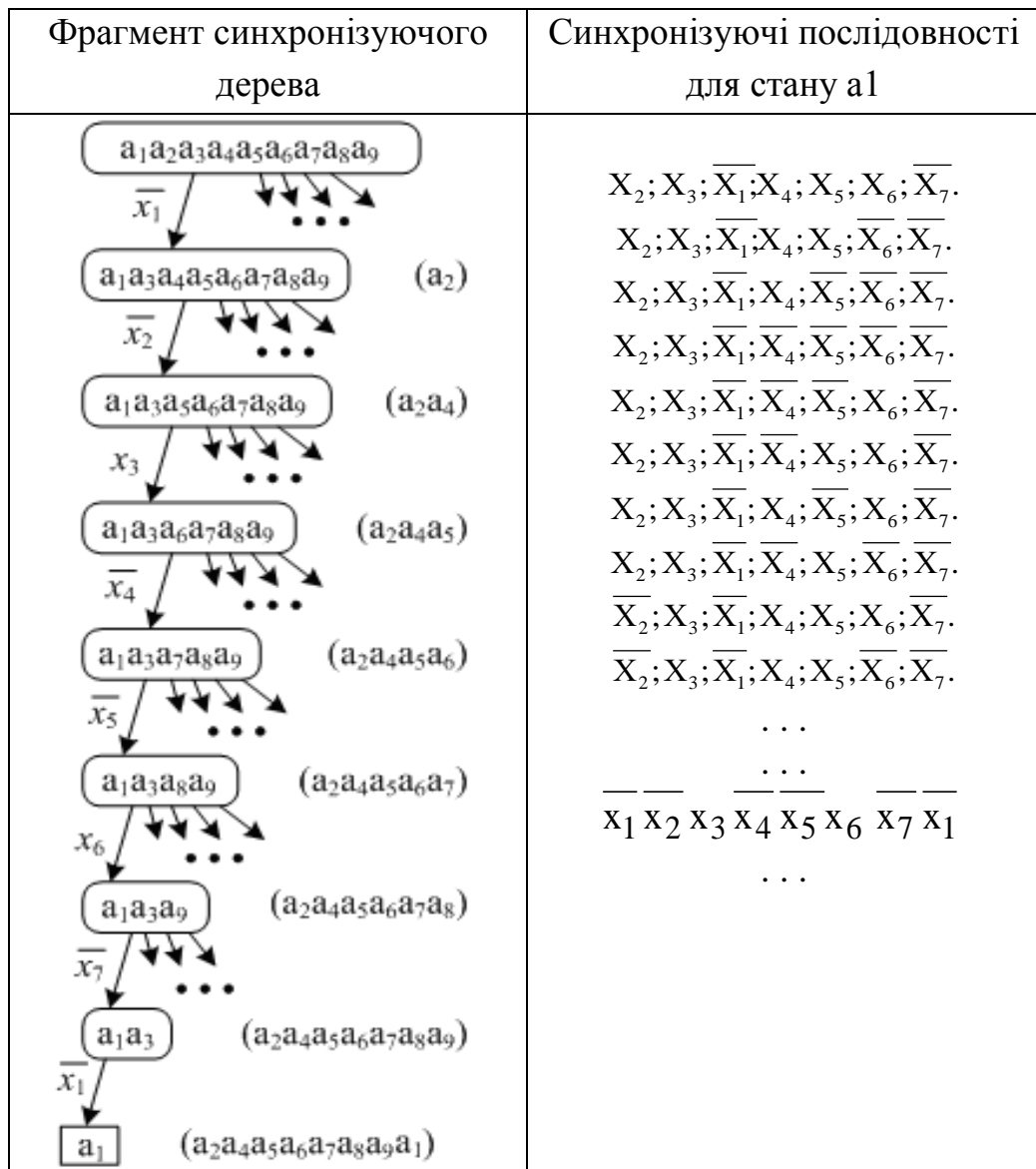


Рисунок 2.8 – Синхронізуючі послідовності для стану a1 КА КП ГРП

Виходячи з синхронізуючого дерева визначимо для аналізу одну синхронізуючу послідовність, а саме, $\overline{\overline{x_1}} \overline{\overline{x_2}} \overline{\overline{x_3}} \overline{\overline{x_4}} \overline{\overline{x_5}} \overline{\overline{x_6}} \overline{\overline{x_7}} \overline{\overline{x_1}}$, або (00100100).

Розглянемо помилкову VHDL-модель з помилкою проектування типу «помилкове призначення нового стану» (фрагмент коду на рис.2.9).

Результати моделювання VHDL-моделі з помилкою призначення нового стану наведені на рис.2.10. На рис. 2.10 показана синхронізуюча послідовність 00100100 (в 16-річному коді 24), що відображено на Waveform, в часовому інтервалі (650-750) ns. Автомат помилково встановлюється в стан a4, а потім примусово повертається в a1.

```

when "0110" =>
    if (X(4)='1') then
        NextState <= A7; Y(6) <= '1';
    elsif (X(4) = '0') then
        NextState <= A4; Y(10) <= '1';
--(замість nextstate <= a1)
    else NextState <= A1;
    end if;

```

Рисунок 2.9 – Фрагмент VHDL-моделі КА з помилковим оператором призначення «a4 замість a1»

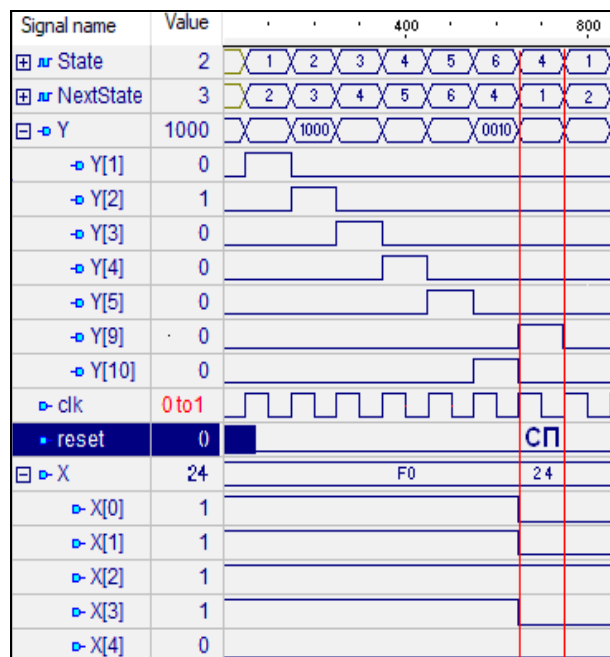


Рисунок 2.10 – Waveform для помилки призначення нового стану «a4 замість a1» з синхронізуючою послідовністю

Таким чином показано, що синхронізуюча послідовність повертає VHDL-модель автомата в початковий стан, і звідси може бути застосована у діагностичному експерименті над реальним цифровим пристроєм. Але для

реального КА може бути визначена така синхронізуюча послідовність, яку фізично реалізувати неможливо на даній системі вхідних датчиків. Наприклад синхропослідовність $\overline{x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_1}$ не може бути реалізована в стані А4, тому що реальний тепловий датчик (X2) не може в одному циклі роботи керуючого автомату попадати в робочій діапазон температур ($x_2=1$), або виходити за його межі ($x_2=0$).

Зазначимо, що в реальних системах логічного управління при діагностуванні КА доцільно використовувати інший підхід, а саме будувати систему імітації вхідних сигналів КА при відключенні самих датчиків.

2.3 Апаратне діагностування керуючих автоматів

При діагностуванні керуючого автомату КП ГРП, що працює з механічними, електричними, термодинамічними датчиками, виникає проблема реалізації вхідних значень електричної напруги, тиску, температури, які виникають при реальній роботі ГРП. Виходячи з цього, пропонується імітувати вихідні значення відповідних датчиків в двійковому алфавіті $\{0, 1\}$ (1 – параметр знаходиться в заданому допустимому інтервалі, 0 – параметр вийшов за межі допустимих значень).

Таким чином, пропонується апаратна реалізація пристрою діагностування (ПД), яке забезпечує виконання всіх переходів по графу керуючого автомата, тобто фактично реалізує його пряму структурну таблицю. Структурна схема об'єкта керування з пристроєм діагностування представлена на рис. 2.11, де: КА – керуючий автомат; ОК – об'єкт керування; ПД – пристрій діагностування; РгД – реєстр даних, куди заноситься черговий варіант послідовності станів обходу графа; У – керуючі сигнали; ХХ – сповіщальні сигнали об'єкта керування (в нашому випадку логічно оброблені показники датчиків); Х – сповіщальні сигнали для керуючого пристрою; УХ – УХ – «імітація» сповіщувальних сигналів пристроєм діагностування; ТМ (test mode) – режим роботи приладу (ТМ = 0 –

робота в режимі керування, $ТМ = 1$ – робота в режимі діагностування).

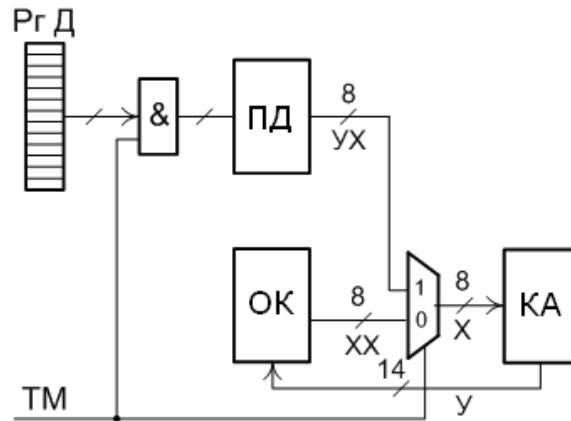


Рисунок 2.11 – Апаратний пристрій діагностування

Суть проведення діагностичного експерименту з використанням ПД полягає в наступному. При підготовці ДЕ будуються шляхи обходу графа переходів КА. При $ТМ = 1$ на вхід ПД надходить послідовність станів, автомата (рис. 2.6), переходи в які необхідно перевірити (порядок опитування відповідних датчиків, який позначається при подачі в КП масивом X), і на виході ПД буде формуватися послідовність сповіщувальних сигналів $УХ$, які «імітують» роботу об'єкта керування (ОК). Сигнали $УХ$ через мультиплексор надходять на КА (в форматі сповіщувальних сигналів X , що надходять з об'єкта управління) та ініціюють його роботу. Керуючий автомат, в свою чергу, формує вихідні сигнали $У$, які надходять на ОК.

Вихідні сигнали $У$ або порівнюються з еталонами (якщо такі є) в режимі відключених датчиків, або, відповідним чином, відображаються на панелі індикації. Якщо послідовність $У$ збігається з еталоном або індикація відображається вірно, то КА працює коректно.

Розглянемо реалізацію алгоритму діагностування. У ньому реалізується модифікована пряма структурна таблиця КА, яка представлена на рис. 2.12.

Схемна реалізація ПД може здійснюватися кількома способами, в залежності від способу кодування вхідної послідовності станів автомата при обході графа переходів.

state	nextstate	X	YX(0-7)
a1	a2	Z=1	10000000
a2	a3	1	00000000
a3	a4	X1=1	01000000
	a1	X1=0	00000000
a4	a5	X2=1	00100000
	a1	X2=0	00000000
a5	a6	X3=1	00010000
	a5	X3=0	00000000
a6	a7	X4=1	00001000
	a1	X4=0	00000000
a7	a8	X5=1	00000100
	a9	X5=0	00000000
a8	a1	X6=0	00000000
	a1	X6=1	00000010
a9	a8	X7=1	00000001
	a1	X7=0	00000000

Рисунок 2.12 – Реалізація алгоритму діагностування

За варіантом 1, послідовність станів автомата, відповідна черговому варіанту обходу станів графа переходів КА кодується бітовим масивом, кожна комірка якого відповідає стану. Якщо біт стану дорівнює 1, то перехід відбувається до наступного стану, позначеного 1. Перехід між станами описаний у вигляді звичайного дво процесного шаблону. Наприклад, якщо $D(2) = 1$ ($D(2)$ відповідає стану a2), то наступним станом і буде a2. Нуль в комірці масиву відповідає поверненню в початковий стан. Схематичне зображення послідовності a1 – a2 – a3 – a4 – a5 – a6 – a1 представлено на рис.2.13. Дане зображення містить повний шлях обходу графу переходів КА.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A1
1	1	1	1	1	1	0	-	-	-

Рисунок 2.13 – Схематичне зображення вхідного масиву КА за варіантом 1

Відсутність бітів «-» означає, що ця ланка не несе ніякої ролі, тому що відбулося повернення в початковий стан ($a_7 = 0$). HDL-модель ПД за варіантом 1 наведена у додатку Е.1

За варіантом 2, ПД приймає на вхід масив 4-х розрядних чисел, кожне з яких є кодом стану КА (наприклад, «0001» відповідає стану a_1 тощо.). Як тільки виявлено код наступного стану, який дорівнює початковому («0001»), то послідовність вважається завершеною і відбувається повернення в початковий стан. Якщо виявлена послідовність не є кінцевою коміркою масиву, то наступні ігноруються. Перехід між станами аналогічний першому варіанту. Схематичне зображення кодування послідовності $a_1 - a_2 - a_3 - a_4 - a_5 - a_6 - a_1$ представлено на рис. 2.14. Воно відображає співвідношення стану до значення в ланці вхідного масиву.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A1
0001	0010	0011	0100	0101	0110	0001	-	-	-

Рисунок 2.14 – Схематичне зображення вхідного масиву і роботи КА за варіантом 2

HDL-модель ПД за варіантом 2 наведена у додатку Е.3.

За варіантом 3 ПД приймає на вхід масив 4-х розрядних чисел, кожне з яких є описом стану. Перший процес відповідає за копіювання вхідного масиву в проміжний і призначення початкових значень сигналам поточного (вихідного – a_1) і подальшого станів з вхідного масиву (для того, щоб виключити подвійне призначення одних і тих же значень). Після призначення сигналам початкових значень, з переднім фронтом синхросигналу відбувається зсув вліво на один елемент масиву та призначення одиниці для уникнення переходів, які не відповідають послідовності. Якщо наступний стан рівний вихідному (початковому – a_1), то послідовність вважається завершеною. Схематичне зображення роботи ПД

показано на рис. 2.15, а його VHDL-модель наведені у додатку Е. HDL-модель ПД за варіантом 3 наведена у додатку Е.5.

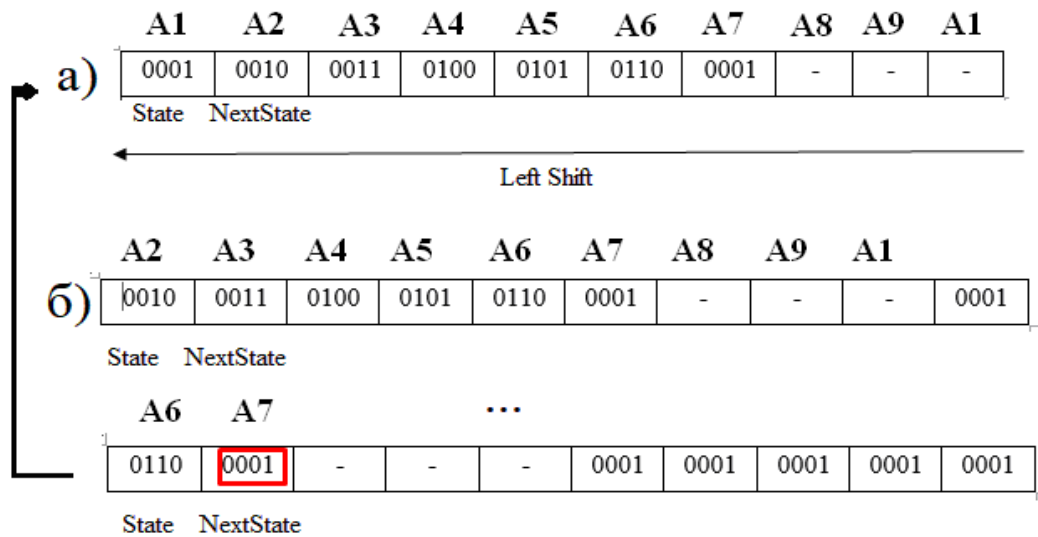


Рисунок 2.15 – Схематичне зображення зсуву вхідного масиву та роботи КА за варіантом 3

Апаратурні витрати по синтезу різних варіантів моделей ПД засобами САПР XILINX ISE в ПЛІС Spartan 3E наведені в таблиці 2.2

Таблиця 2.2 – Результати синтезу ПД (Device utilization summary)

Selected Devic: 3s500efg320-5	Варіант 1	Варіант 2	Варіант 3
Number of Slices	13 out of 4656	27 out of 4656	23 out of 4656
Number of Slice Flip Flops	16 out of 9312	16 out of 9312	35 out of 9312
Number of 4 input LUTs:	15 out of 9312	21 out of 9312	17 out of 9312
Number of IOs	20	50	10
Number of bonded IOBs:	17 out of 232	39 out of 232	10
IOB Flip Flops:	7	28	10
Number of GCLKs	1	1	1

Аналіз результатів синтезу різних варіантів схемних реалізацій ПД

показав наступне.

1. Апаратні витрати на реалізацію ПД за першим варіантом мінімальні, але при такому способі кодування послідовності станів автомата виникають проблеми, якщо граф переходів є мультиграфом, тобто між парою вершин є більше однієї дуги, переходи за якими визначаються різними сповіщувальними сигналами (для данного автомата це пара станів a_8 – a_1).

2. В схемній реалізації ПД за варіантом 3 є приблизно в 2 рази більше тригерів (Slice Flip Flops), ніж в інших варіантах реалізації. Це обумовлено тим, що в цьому варіанті вхідний масив `c_Test_sequence` копіюється в регістр DD, що фактично імітує роботу з зовнішньою енергонезалежною пам'яттю.

3. З точки зору масштабованості ПД щодо числа станів автомата, третій варіант схемної реалізації є кращим, тому що зі збільшенням довжини послідовності для обходу графа автомата апаратні витрати в цьому варіанті ростуть в незначній кількості, чого не можна сказати про варіант 2.

4. Апаратні витрати на ПД при різних варіантах схемної реалізації можна порівняти з апаратними витратами на КА (табл.2.1), що засвідчує приблизно рівні апаратні витрати на КА і ПД (не враховуючи модель енергонезалежної пам'яті у варіанті 3). Це підтверджує правильність обраної методики побудови апаратного пристрою діагностуванняБ бо не порушується принцип перевищення апаратних витрат над дублюванням.

З'єднання КП і ПД являє собою певну систему управління і діагностування пристрою або, навіть, окремої системи. КА може працювати окремо від ПД, займаючись контролем роботи системи або пристрою, у вигляді подачі вихідних сигналів, що ініціюють виконання операцій. Використання повної системи управління і діагностування є тестовим режимом перевірки поведінки пристрою або системи.

У лістингу 2.2 представлений HDL-код, що описує зазначену систему. Це структурна модель, що складається з двох компонентів. Компонент КА пов'язаний з компонентом ПД регістром X. Це вихідний масив ПД і вхідний масив КА для генерації значень перевірки умов.

Лістинг 2.2 – Опис структурної моделі КА та ПД

```

-- Control and diagnostic system
-- Contains Diagnostic and Control devices
-- Shows system working process and can execute test mode

library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity Workstation is

    port(
        clk,reset : in STD_LOGIC;
        Y:out STD_LOGIC_VECTOR(1 to 14)
    );
end Workstation;

architecture Workstation of Workstation is

    component GAS_FSM is
        port(
            clk: in STD_LOGIC;
            reset: in STD_LOGIC;
            X: in STD_LOGIC_VECTOR (0 to 7);
            Y: out STD_LOGIC_VECTOR (1 to 14)
        );
    end component GAS_FSM;

    component GAS_Diagnostic_Device is
        port(
            clk: in STD_LOGIC;
            reset: in STD_LOGIC;
            Yx: out STD_LOGIC_VECTOR (0 to 7)
        );
    end component GAS_Diagnostic_Device;

    signal Yx:STD_LOGIC_VECTOR(0 to 7);
begin

    UUT0:GAS_Diagnostic_Device
        port map(clk=>clk,reset=>reset,Yx=>Yx);

    UUT2:GAS_FSM
        port map(clk=>clk,reset=>reset,X=>Yx,Y=>Y);
end Workstation;

```

Побудовані VHDL-моделі керуючого і діагностичного автоматів верифіковані інструментальними засобами системи моделювання Active-HDL. Результати верифікації представлені на часовій діаграмі (waveform), на якій візуально відображається закон функціонування зазначених автоматів. На рис. 2.16 представлений результат моделювання роботи ПД по реалізації варіанту обходу графа a1 – a2 – a3 – a4 – a5 – a6 – a1. Результат порівняння роботи КА і ПД представлений на рис.2.17, а на рис.2.18 представлена повна Waveform з реалізацією алгоритму діагностування, що підтверджує їх повну ідентичність.

Signal name	Value		200	400	600				
State	Сигнали ПД	6	U	1	2	3	4	5	6
NextState		1	U	2	3	4	5	6	1
Yx	Виход ПД і вход КА	00	00	80	00	40	20	10	00
X		00	00	80	00	40	20	10	00
Y	Результат	0010		2000	1000	0800	0400	0200	0010
State	Сигнали КА	6	U	1	2	3	4	5	6
NextState		1	U	2	3	4	5	6	1

Рисунок 2.16 – Детальний опис сигналів структурної моделі

Signal name	Value		200	400	600			
State	6	U	1	2	3	4	5	6
NextState	1	U	2	3	4	5	6	1
Yx	00	00	80	00	40	20	10	00
Yx[0]	0							
Yx[1]	0							
Yx[2]	0							
Yx[3]	0							
Yx[4]	0							
Yx[5]	0							
Yx[6]	0							
Yx[7]	0							
X	00	00	80	00	40	20	10	00
X[0]	0							
X[1]	0							
X[2]	0							
X[3]	0							
X[4]	0							
X[5]	0							
X[6]	0							
X[7]	0							
Y	0010		2000	1000	0800	0400	0200	0010
State	6	U	1	2	3	4	5	6

Рисунок 2.17 – Порівняння виходу ПД та входу КА

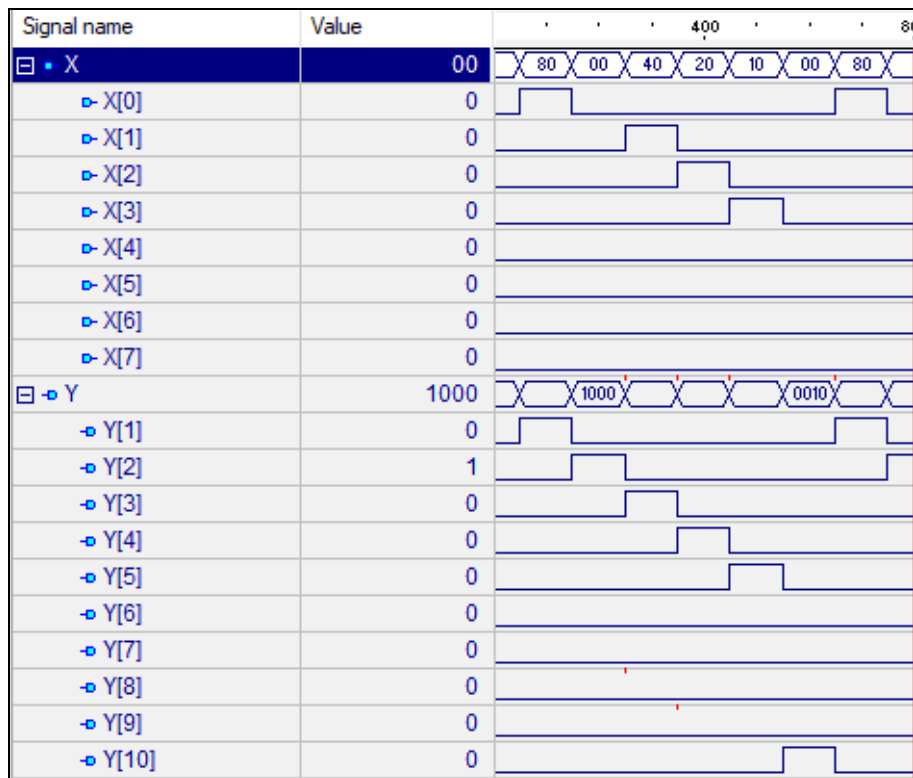


Рисунок 2.18 – Результат роботи пристроїв керування та діагностування (послідовність a1-a2-a3-a4-a5-a6-a1)

Загальні апаратні витрати по синтезу моделі КА и ПД засобами САПР XILINX ISE в ПЛІС Spartan 3Е наведені в таблиці 2.3

Таблиця 2.3 – Результати синтезу по моделі КА и ПД

Selected Device : 3s500efg320-5	ПЛІС Spartan 3Е
Number of Slices	34 out of 4656
Number of Slice Flip Flops	44 out of 9312
Number of 4 input LUTs:	38 out of 9312
Number of IOs	16
Number of bonded IOBs:	17 out of 232
Number used as Shift registers: 1	1
Number of GCLKs	1

2.4 Висновки до розділу 2

1. Розглянуті функції та місце контрольного пункту (КП) у складі газорегуляторного пункту (ГРП) Наведено алгоритм роботи КП ГРП у формі граф-схеми алгоритму, граф переходів керуючого автомату, створена модель керуючого автомату на мові опису апаратури VHDL. Проведена її верифікація VHDL-моделі та автоматизований синтез пристрою керування. Оцінені апаратурні витрати на керуючий автомат.

2. Показані задачі діагностування, як засобу підтримання надійної роботи критичної системи безперебійного газопостачання. Для діагностування керуючого автомату запропоновано проводити «неруйнівний» діагностичний експеримент з обходу всіх вершин (дуг) графу переходів автомату з використанням синхронізуючих послідовностей. Отримали подальший розвиток процедури використання синхронізуючих послідовностей при проведенні «неруйнівного» діагностичного експерименту, який забезпечує повернення автомату в початковий стан навіть при наявності в ньому несправностей. Наведений приклад побудови синхронізуючої послідовності та діагностичний експеримент з її використанням. Моделювання VHDL-моделі при проведенні діагностичного експерименту здійснювалося в середовищі Active-HDL.

3. Для автоматичного проведення діагностичного експерименту з обходу всіх станів керуючого автомату без відключення його від основного режиму роботи на тривалий час запропоновано використання вбудованого пристрою діагностування. Це дозволило проводити процедуру діагностування керуючого автомата у віддаленому режимі навіть без присутності кваліфікованого персоналу. Проведена оцінка апаратурних витрат на вбудований пристрій діагностування і показана доцільність застосування в ньому зсувного регістру для збереження послідовності станів.

РОЗДІЛ 3

ТЕСТОПРИДАТНЕ ПРОЕКТУВАННЯ КЕРУЮЧИХ АВТОМАТІВ

В третьому розділі розглянуті методи тестопридатного проектування керуючих автоматів за рахунок розширення вхідного алфавіту автомату шляхом введення додаткового стовпця в таблицю переходів-виходів автомата, що суттєво скорочує довжину та час діагностичного експерименту за рахунок можливості встановлювати автомат в довільний стан.

В ході проведення досліджень мають бути вирішені такі завдання:

1) розробити методи підвищення тестопридатності моделей кінцевих автоматів за рахунок розширення вхідного алфавіту шляхом введення додаткового стовпця в таблицю переходів-виходів, що дозволить забезпечити встановлення керуючого автомата у довільний стан за $(n-1)$ тактів, де n – число станів автомата;

2) проаналізувати додаткові апаратурні витрати, які виникають при розширенні вхідного алфавіту, по результатам автоматизованого синтезу легкотестованих моделей керуючих автоматів, наданих мовами опису апаратури;

3) удосконалити методи оцінки тестопридатності графових моделей керуючих автоматів на основі розрахунку досяжності вершин графу переходів;

4) розробити методи проведення неруйнівного діагностичного експерименту в графових моделях керуючих автоматів за рахунок організації гамільтонових циклів в графі переходів.

При проектуванні тестопридатних керуючих автоматів апаратурну надлишковість, що забезпечує легкотестованість, доцільно вносити ще на початковому етапі проектування, тобто на етапі побудові HDL-моделей автоматних пристроїв, що проектуються.

Основні результати досліджень опубліковані в роботах [117, 118, 121, 125].

3.1. Побудова легкотестованих автоматів шляхом розширення вхідного алфавіту

Розглянемо метод розширення автоматної моделі об'єкта діагностування (ОД) з метою перетворення його в легкотестований шляхом додавання одного вхідного символу, що забезпечує побудову повного безумовного перевіряючого експерименту, довжина якого є мінімальною для прийнятих обмежень. У моделі, що розглядається, автомат передбачається кінцевим, синхронним, детермінованим, сильнопов'язаним і повністю визначеним. [111].

Перевіряючим (діагностичним) експериментом будемо називати вхід-вихідну послідовність, яка ідентифікує еквівалентність автомата, що перевіряється, справному при заданому обмеженні на клас несправностей. Клас несправностей, що виявляється, включає множину стійких несправностей, що довільно змінюють таблицю переходів виходів (ТПВ) справного автомата, при обмеженні, що несправність не збільшує числа станів автомата. Таким чином, при побудові ДЕ використовується функціональна модель несправностей переходів автомата, яка покриває клас кратних константних несправностей, підклас коротких замикань, переплутувань і інверсій.

Легкотестованим будемо називати кінцевий автомат, для якого можна побудувати ДЕ мінімальної довжини шляхом забезпечення встановлення КА у довільний стан за $(n-1)$ тактів, де n – число станів автомата. Для легкотестованого КА існує безумовний діагностичний експеримент, який має довжину, прийняту для заданих умов перевірки його справності й алгоритмічно просто реалізований. Властивість алгоритмічної простоти реалізації перевіряючого експерименту дозволяє скоротити вартість зовнішніх засобів діагностування та здійснити перевірку справності вбудованими засобами діагностування, що особливо актуально з розвитком виробництва і застосування SoC. Як впливає з аналізу методів, побудова

повних перевіряючих експериментів з автоматами, легкотестованим є мінімальний, сильнозв'язний автомат, який має відмінну послідовність мінімальної довжини, синхронізуючі і переводячі послідовності. Якщо пам'ять автомата реалізована на тригерних елементах, то з урахуванням прийнятого обмеження на клас виявлених несправностей, які не збільшують числа станів автомата, ймовірність появи такої несправності незначна, коли число станів точно дорівнює ступеню 2, тобто $n=2^k$, де k – число використовуваних тригерів. Якщо $n \neq 2^k$, то такі несправності дуже ймовірні. Виходячи з цього, бажано, щоб вихідний автомат мав усі $2^k = n$ станів.

Процедура перетворення автомата, представлена нижче, заснована на властивостях лічильникових і зсувно-реєстрових послідовностей. Вперше Мак Коннелом [113] було запропоновано використовувати лічильникові функції переходів і виходів автомата для перетворень його в легкотестований. Ми будемо розглядати функції переходів і виходів окремо. З цією метою введемо таке визначення.

Нехай в автоматі $A = (X, Y, A, \delta, \lambda)$ з множиною станів $A = \{z_1\}$, $i = \overline{1, n}$, є вхідний символ $x_0 \in X$, додаток якого породжує функцію переходів:

$$\delta(z_i, x_0) = z_{(i+1)} \bmod n, \quad \forall i = \overline{1, n}, \quad (3.1)$$

Нехай в автоматі $A = (X, Y, Z, \delta, \lambda)$, із множиною станів $Z = \{z_1\}$, $i = \overline{1, n}$, є вхідний символ $x_0 \in X$, додаток якого породжує функцію переходів

$$\delta(z_i, x_0) = z_{(i+1)} \bmod n \quad \forall i = \overline{1, n}, \quad (3.2)$$

Функцію переходів виду (3.2) будемо називати лічильниковою функцією переходів (ЛФП). Додавання стовпця в ТПВ довільного автомата забезпечує перетворення його в сильнозв'язний автомат, що має для кожного i -го стану множину послідовностей, що переводяться, у вигляді (3.4)

$$T(z_i, z_{i+1}) = x_0, T(z_i, z_{i+1}) = x_0, \dots, T(z_i, z_{i+k}) = x_0^{(k)}, \dots \quad (3.4)$$

де $x_0^{(k)}$ – конкатенація вхідного символу x_0 k – разів.

Якщо автомат $A = (X, Y, A, \delta, \lambda)$, $Y = \{y_j\}, j = \overline{1, r}$, має вхідний символ x_0 , якому відповідала б $\lambda(z_j, x_0) = y_j^i$, де верхній індекс «i» відповідає індексу попереднього стану автомата, то додаток до A вхідної послідовності $x_0^{(k)}$ довжиною в k - символів викликає поява вихідної послідовності, а верхні індекси підсумовуються по mod n .

$$\lambda(z_j, x_0^{(k)}) = y_j^i, y_j^{(i+1)}, \dots, y_j^{(i+k)}, j = \overline{1, r}, \quad (3.5)$$

Не порушуючи спільності міркувань, розглянемо ТПВ довільного автомата A_1 , що має $n = 8$ станів. Розширимо ТПВ автомата додаванням шпальти x_0 (рис.3.2), в якому функція переходів – СФП згідно з (3.3), а функція виходів (3.5) кодується послідовністю $P_g(3)$, взятої з таблиці 1. Як видно з розпізнавального дерева на рис. 3.1, автомат A_1 має однорідну відмінну послідовність мінімальної довжини $l = (x_0) = 3$.

$z(t)$	x_1	...	x_m	x_0	
z_0	.	.	.	$z_1, 1$	$\{z_0 \div z_7\}$
z_1	.	.	.	$z_2, 1$	$ x_0$
z_2	.	.	.	$z_3, 1$	$\{z_1, z_2, z_3, z_5\} \{z_0, z_4, z_6, z_7\}$
z_3	.	.	.	$z_4, 0$	$ x_0$
z_4	.	.	.	$z_5, 1$	$\{z_0\}, \{z_1\}, \dots, \{z_7\}$
z_5	.	.	.	$z_6, 0$	
z_6	.	.	.	$z_7, 0$	
z_7	.	.	.	$z_0, 0$	

Рисунок 3.1 - ТПВ та розрізняюче дерево автомата A_1

У загальному випадку для автомата з n станами справедливо наступне твердження.

Нехай в автоматі $A = (X, Y, A, \delta, \lambda)$, у якого $z = \{z_i\}, i = \overline{1, n}, y = \{y_j\}, y_j \in \{0, 1\}$ є вхідний символ, який визначає СФП і де $P_g(n)$ послідовність з n двійкових символів. Якщо Π -відображення послідовності $P_g(n)$, що кодує функцію виходів автомата, у послідовність вершин підграфа переходів $G_n \subseteq G_2^k$ k -розрядного ЗР є його гамільтоновим циклом, то A має однорідну відмінну послідовність $x_0 = x_0^{(k)}$ довжиною $k = \lceil \log_2 n \rceil$.

Дане твердження легко узагальнити на клас автоматів, що мають n станів і « m » вихідних символів, а, отже, $r = \lceil \log_2 m \rceil$ вихідних змінних. Ймовірно, для такого автомата довжина мінімальної відмінної послідовності дорівнює $\left\lceil \frac{\log_2 n}{\log_2 m} \right\rceil$.

Використовуючи r вихідних змінних $\{Y_1, Y_2, \dots, Y_r\}$, закодуємо функцію виходів у стовпці ТПВ x_0 у такий спосіб. Нехай множині станів $z(t) = \{z_1, z_2, \dots, z_n\}$ однозначно відповідає множина вихідних символів $\{y_1, y_2, \dots, y_n\}$ на вихідному полюсі Y_1 , при вхідному символі x_0 . Закодуємо функції виходів інших змінних $(r-1)$ наступним чином:

$$\lambda(t) = \begin{vmatrix} Y_1 & Y_2 & Y_3 & \dots & Y_r \\ y_1 & y_2 & y_3 & \dots & y_r \\ y_2 & y_3 & y_4 & \dots & y_{r+1} \\ y_3 & y_4 & y_5 & \dots & y_{r+2} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ y_n & y_1 & y_2 & \dots & y_{r-1} \end{vmatrix} \quad (3.6)$$

У (3.6) вихідні змінні Y_2, Y_3, \dots, Y_r кодуються послідовностями, які виходять циклічним зрушенням вліво на один символ y_i кожної попередньої послідовності. Якщо $\{y_1, y_2, \dots, y_n\}$ – послідовність, що породжує гамільтонів цикл в підграфі G_n . Це означає, що при $m=n$ кожен рядок матриці (3.6) буде

відповідати двійковому коду вершини підграфу переходів $G_n \subseteq G_2^k$ k -розрядного ЗР і відрізняти їхні капітали автомата один від одного за один крок додатка вхідного символу x_0 . Для $m < n$, число додатків x_0 , що утворюють відмінну послідовність, визначиться відношенням $\left\lceil \frac{\log_2 n}{\log_2 m} \right\rceil$.

Розглянемо процедуру побудови повної перевіряючої послідовності для автоматів, що мають в ТПВ стовпчик x_0 , кодований за запропонованою вище методикою.

Нехай автомат має n станів, m вхідних символів, що не виключають символ x_0 , і « k » вихідних символів. Стовпець x_0 ТПВ забезпечує автомат відмінною послідовністю x_0 мінімальної довжини $L_0 = \left\lceil \frac{\log_2 n}{\log_2 k} \right\rceil$, а лічильникова функція переходів – послідовностями, що переводять $l(T(z_i, z_{i+n-1})) = (n-1)$.

Повна довжина перевіряючої послідовності автомата, перетвореного на основі запропонованого методу, визначається шляхом підсумовування довжин всіх чотирьох фаз експерименту.

В результаті отримаємо:

$$l(T) \leq \left(\frac{mn}{2} + 2 \right) \cdot (n + 2L_0 + 3), L_0 = \left\lceil \frac{\log_2 n}{\log_2 k} \right\rceil \quad (3.7)$$

Незважаючи на існування в автоматі розрізняючої послідовності мінімальної довжини, довжина повної перевіряючої послідовності зростає в квадратичній залежності від числа станів. Ця залежність визначається сумарною довжиною послідовностей, що переводять. Однак можна показати, що з урахуванням прийнятих обмежень запропонований метод перетворення автомата в легкотестований є оптимальним, а отримана оцінка верхньої межі довжини (8) є мінімальною для довільного автомата, у якого процедура перетворення в легкотестований передбачає введення одного додаткового вхідного символу.

За початковим припущенням вихідний автомат не обов'язково мінімальний і сильнозв'язний. Додаючи один стовпець x_0 в ТПВ вихідного автомата, необхідно забезпечити його мінімальність, сильнозв'язність і існування однорідної відмінної послідовності.

Для цього необхідно і достатньо, щоб стовпець x_0 являв собою мінімальний підавтомат з n станами. Так як стовпець x_0 містить всі n станів, то властивість сильнозв'язності може бути забезпечено тільки лічильниковою функцією переходів або будь-яким її ізоморфним відображенням на безлічі всіх станів, що не містить приватних циклів довжиною менше n . Сумарна довжина безлічі переводячих послідовностей для такої функції переходів не перевищує $\frac{n(n-1)}{2}$ вхідних символів.

Кодування функції виходів в стовпці x_0 послідовностями $P_g(1)$, як було обгрунтовано вище, забезпечує існування в автоматі однорідної відмінної послідовності мінімальної довжини. Таким чином, при наявності в автоматі синхронізуючої послідовності мінімальної довжини (вхід x_c – скидання в початковий стан), побудований експеримент має мінімальну довжину повної послідовності, що перевіряє.

Розширення ТПВ автомата шляхом додавання стовпця x_0 у загальному випадку реалізуючого довільну функцію переходів-виходів, підвищує керованість станів автомата і призводить до перетворення структури ОД до стандартного вигляду Scan Path.

3.2 Проектування HDL-моделей тестопридатних автоматів

У сучасному циклі автоматизованого проектування цифрових пристроїв (ЦП) специфікація на проектування пристрою задається, як правило, у вигляді алгоритму функціонування, представленого на одній з мов опису апаратури. Для такого способу подання ЦП застосування структурних методів діагностування, орієнтованих на виявлення константних

несправностей, стає мало ефективним. З іншого боку, застосовувати функціональні методи досить складно з огляду на розмірності сучасних проєктованих пристроїв.

Одним з поширених способів вихідного опису спеціалізованих ЦП обробки даних та управління є кінцевий автомат, а формою його подання таблиця переходів-виходів (і побудований на її основі граф переходів автомата). Зважаючи на складність діагностичних експериментів з автоматами, були запропоновані різні методи забезпечення тестопридатності, тобто модифікації автоматних моделей ЦП, що передбачають введення апаратної надмірності, як на структурному рівні шляхом введення додаткових входів і виходів для забезпечення простоти проведення ДЕ, так і на функціональному рівні за рахунок внесення доповнень та змін у функціональний опис автомата, а саме, в його ТПВ. Такий підхід дозволяє зробити тестопридатний автомат легкотестованим, тобто таким, для якого завдання забезпечення тестового діагностування вирішуються максимально просто в межах встановлених витрат на проєктування.

При високорівневому проєктуванні цифрових пристроїв управління на основі кінцевих автоматів формою подання специфікації проєктованого пристрою є таблиця переходів-виходів (state table) або граф переходів автомата (state diagram). Одним із способів опису моделей ЦП в формі кінцевих автоматів на мові VHDL є автоматний шаблон, тобто спосіб опису моделей керуючих кінцевих автоматів, специфікація на які задана у вигляді ТПВ або графа переходів. Це спеціальна структура VHDL-моделі, в якій функції переходів і виходів виділені в окремі процеси (процес), а призначення нового стану здійснюється в спеціальному процесі, пов'язаному з синхронізацією. При проєктуванні тестопридатних керуючих автоматів апаратну надмірність, що забезпечує легкотестованість, доцільно вносити ще на початковому етапі проєктування, тобто при побудові HDL-моделей проєктованих пристроїв. Враховуючи вищезазначене, легкотестованим будемо називати кінцевий автомат, для якого можна побудувати

діагностичний експеримент мінімальної довжини шляхом забезпечення установки автомата у будь-який стан за мінімальне число тактів.

Таким чином, актуальною є задача розробки процедур автоматизації проектування легкотестованих кінцевих автоматів, представлених на мовах опису апаратури, порівняння додаткових витрат апаратури при структурних і функціональних методах підвищення тестопридатності на основі порівняння результатів синтезу тестопридатності HDL-моделей засобами САПР ПЛІС

Схема є тестопридатною, якщо процедура генерації множини тестових наборів, оцінка їх ефективності та реалізація тестового діагностування можуть бути виконані за умови дотримання у встановлених межах фінансових витрат, витрат часу і значень показників, що характеризують пристосованість схеми до виявлення несправностей, пошуку місця несправностей та реалізації тестового діагностування [85].

Експериментом над автоматом будемо називати процес додавання вхідних послідовностей, спостереження відповідних їм вихідних послідовностей та виведення висновків, заснованих на цих спостереженнях. Залежно від мети експерименту розрізняються експерименти по ідентифікації станів автомата, ідентифікації вхідної послідовності автомата та ідентифікації автомата з n станами, що відрізняється від всіх інших автоматів з таким же числом станів. З метою виявлення помилок проектування в моделях керуючих автоматів, представлених ТПВ або графом переходів автомата, застосовується діагностичний експеримент, пов'язаний з обходом всіх вершин графа. З цієї точки зору легкотестованим є автомат, який може бути встановлений в будь-який стан не більше, ніж за n тактів, де n – число станів автомата без використання синхронізуючих послідовностей [8].

Якщо розглядати як об'єкт діагностування HDL-моделі кінцевих автоматів в формі автоматного шаблону, то для побудови на їх основі легкотестованих автоматів розглянуті моделі повинні відповідати таким умовам:

- в моделі повинен бути спеціальний механізм переводу автомата в

режим тестування і, навпаки, на будь-якому такті роботи автомата;

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

– легкотестований автомат повинен будуватися в автоматизованому режимі засобами САПР.

Як приклад перетворення HDL-моделей керуючих автоматів розглянемо моделі автоматів Мура та Мілі на мові VHDL. На рис. 3.2 представлені таблиці переходів-виходів вказаних автоматів.

a/y	1	x1	\bar{x}_1	x2	\bar{x}_2x_3	$\bar{x}_2\bar{x}_3$
a1	a2					
a2	y1	a4	a3			
a3	y2	a5				
a4	y3			a3	a4	a5
a5	y4	a1				

а)

a	1	x ₁	\bar{x}_1	x ₂	\bar{x}_2x_3	$\bar{x}_2\bar{x}_3$
a ₁	a ₂ / y ₁ ,y ₂					
a ₂		a ₃ /y ₃	a ₄ /y ₄			
a ₃	a ₅ /-					
a ₄				a ₃ /y ₄ ,y ₅	a ₄ /-	a ₅ /y ₃
a ₅	a ₁ / y ₅					

б)

Рисунок 3.2 – ТПВ автоматів Мура (а) та Мілі (б)

Надалі (якщо не обумовлено інше) будемо розглядати автомати, робота яких тактується сигналами від генератора синхронізуючих імпульсів (CLK), тобто в автоматі крім входів (x_1, \dots, x_n) є принаймні ще один вхід CLK, за яким надходить синхронізуюча послідовність. Тому вхідним сигналом на переході (a_m, a_s) , відповідним шляху $a_m X(a_m, a_s) - Y(a_m, a_s) a_s$, буде не $X(a_m, a_s)$, а кон'юнкція $CLK \cdot X(a_m, a_s)$. Таким чином, при $X(a_m, a_s) = 1$ (безумовний перехід) автомат працює тільки від синхроімпульсу. Але при побудові графа переходів автомата символ CLK дуг графа переходів, як правило, не приписується.

3.2.1 Тестопридатна структура з використанням регістру зсуву

Розглянемо структурний спосіб організації сканованого шляху в HDL-моделях кінцевих автоматів шляхом безпосереднього введення зсувного регістру в HDL-модель вручну. Але запропонована модель дещо відрізняється від стандартного автоматного шаблону, тому що коди станів автомата при цьому мають задаватися примусово.

На рис. 3.3 наведено фрагмент VHDL-моделі автомата Мілі в формі автоматного шаблону зі зсувним регістром в частині, що запам'ятовує.

```
architecture FSM_MX of FSM_Mealy_MX is
    signal State, NextState: STD_LOGIC_vector (2 downto 0);
    signal a1: STD_LOGIC_vector (2 downto 0):="001";
    signal a2: STD_LOGIC_vector (2 downto 0):="010";
    signal a3: STD_LOGIC_vector (2 downto 0):="011";
    signal a4: STD_LOGIC_vector (2 downto 0):="100";
    signal a5: STD_LOGIC_vector (2 downto 0):="101";
begin
    Sreg0_CurrentState: process (Clk, Reset)
    begin
        if Reset='1' then State <= a1;
        elsif Clk'event and Clk = '1' then
            if A='1' then State <= NextState;
            else State <= State(1 downto 0) & TDI;
            end if;
        end if;
    end process;
end architecture;
```

Рисунок 3.3 – Фрагмент архітектури VHDL-моделі автомата Мілі з зсувними регістром (FSM_Mealy_MX)

Сигнал $A = 1$ задає режим стандартного функціонування автомата, $A = 0$ – режим зсуву в регістрі з зовнішнього входу TDI (test data input) в сторону старших розрядів. Стани автомата в даній моделі кодуються примусово за номером стану (при декларації внутрішніх сигналів станів автомата) з використанням мінімального числа тригерів (в даному випадку 3) для зберігання коду стану. Така організація моделі дає можливість

записувати ззовні з входу TDI (test data input) код потрібного стану за мінімальне число тактів, яке визначається числом розрядів зсувного регістру.

На рис. 3.4 приведена часова діаграма (waveform) моделювання роботи автомата Мілі з зсувним регістром в запам'ятовуючій частини та зовнішнім входом TDI (FSM_Mealy_MX). Сигнал State наведено на діаграмі в десятковому вигляді. При $A = 1$ (від 0 нс до 500 нс) автомат працює в нормальному режимі, при $A = 0$ (від 500 нс до 2000 нс) – у зсувному режимі, при цьому значення TDI записується в старші розряди.

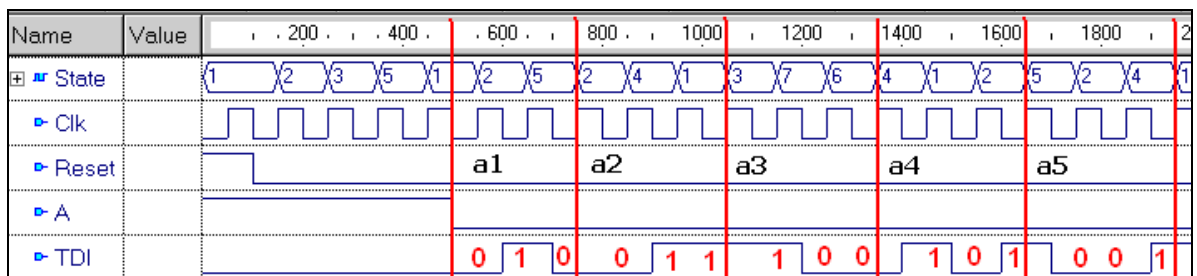


Рисунок 3.4 – Часова діаграма роботи автомата Мілі з зсувним регістром (FSM_Mealy_MX)

На рис. 3.5 наведено фрагмент схеми автомата Мілі з зсувним регістром і зовнішнім входом TDI (FSM_MX) після синтезу в системі Xilinx ISE.

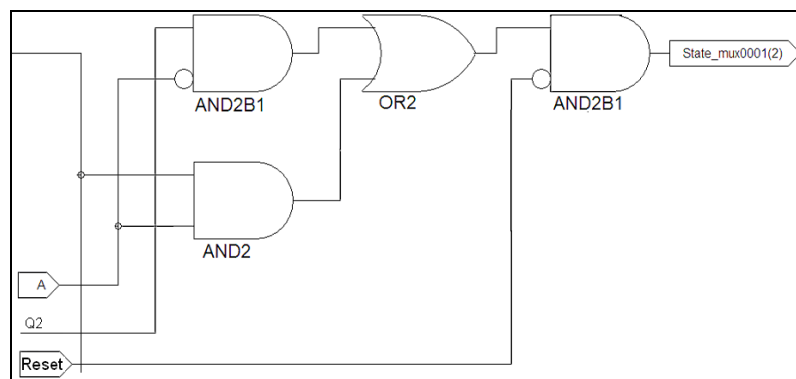


Рисунок 3.5 – Фрагмент схеми автомата Мілі з зсувним регістром (FSM_Mealy_MX)

Повне рівняння для функції збудження тригера D2 буде:
 $D_2 = \{[(x_3 \vee \overline{x_3}) y_3 \overline{x_2} \vee (x_1 y_1 \vee y_2)] A \vee Q_2 \overline{A}\} \overline{Reset}$. Даний фрагмент схеми, що синтезується, фактично є мультиплексором щодо керуючого сигналу A, що повністю відповідає структурі схеми зі скануванням шляху.

З одного боку запропонована методологія структурного перетворення автоматів у легкотестовані за рахунок введення зсувних регістрів вирішує поставлене завдання, але з іншого боку має ряд недоліків.

1. Поява додаткового входу TDI є корисним при організації структури Boundary Scan, але створює додаткові труднощі при організації роботи автомата, тому що в стандартному режимі роботи керуючого автомата даного входу не передбачено.

2. Розімкнутий зсувний регістр створює додаткові складності для організації гамільтонового циклу при скануванні автомата.

3. Використання зсувного регістру при довільному кодуванні станів автомата знижує швидкодію в режимі сканування.

Таким чином, запропонована методика не вирішує всіх поставлених завдань, потрібно інше рішення. Воно знаходиться в області функціонального перетворення ТПВ автомата шляхом розширення вхідного алфавіту для забезпечення тестопридатності.

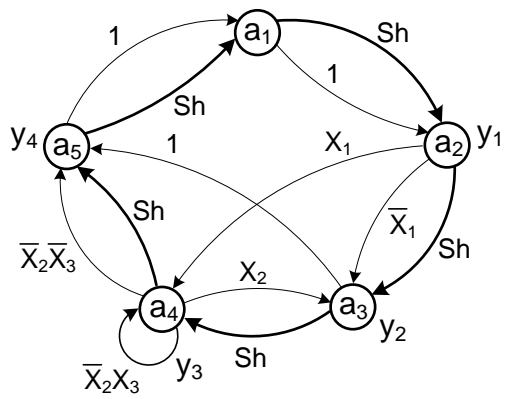
3.2.2 Тестопридатна структура із скануванням станів

При функціональному підході до проектування (аналізу) цифрових пристроїв, заданих моделлю кінцевого автомата $W = \langle X, A, Y, \delta, \lambda \rangle$ підвищення тестопридатності автомата можливо тільки за рахунок розширення вхідного алфавіту X, алфавіту станів A або вихідного алфавіту Y [68]. З одного боку це призводить до внесення в схему пристрою, що проектується, апаратурної надлишковості, а з іншого боку забезпечує збереження алгоритму функціонування автомата, заданого, як правило, ТПВ або його графом переходів.

У підрозділі 3.1 запропоновано та теоретично обґрунтовано введення додаткового стовпця (символу) в таблицю переходів-виходів автомата, що забезпечує для цього символу функцію переходів графа переходів зсувного регістру та дозволяє встановити автомат в будь-який заданий стан. На підставі зазначених теоретичних положень розширимо ТПВ автоматів Мура (рис.3.2 а) та Мілі (рис. 3.2 б) додаванням стовпця Sh (Shift). При Sh = 1 автомат працює в режимі установки в будь-який заданий стан, а при Sh = 0 автомат реалізує заданий алгоритм. Розширена ТПВ та модифікований граф переходів автомата Мура показана на рис.3.6, а автомата Мілі на рис. 3.7.

a/y	1	x1	\bar{x}_1	x2	\bar{x}_2x_3	$\bar{x}_2\bar{x}_3$	Sh
a1	a2						a2
a2	y1	a4	a3				a3
a3	y2	a5					a4
a4	y3			a3	a4		a5
a5	y4	a1					a1

а)

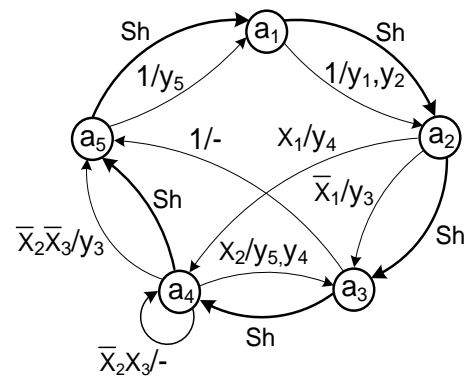


б)

Рисунок 3.6 – Розширена ТПВ та модифікований граф переходів автомата Мура з сигналом Sh

a	1	x1	\bar{x}_1	x2	\bar{x}_2x_3	$\bar{x}_2\bar{x}_3$	Sh
a1	a2/ y1,y2						a2/-
a2		a3/ y3	a4/ y4				a3/-
a3	a5/-						a4/-
a4				a3/y4, y5	a4/ -	a5/ y3	a5/-
a5	a1/ y5						a1/-

а)



б)

Рисунок 3.7 – Розширена ТПВ та модифікований граф переходів автомата Мілі з сигналом Sh

Для перевірки ефективності запропонованого методу підвищення тестопридатності автомата за рахунок розширення вхідного алфавіту та введення додаткового стовпця у ТПВ розглянемо модифікацію VHDL-моделей автоматів Мілі та Мура.

На рис. 3.8 представлений фрагмент архітектури VHDL-моделі легкотестованого автомата Мура з додатковим сигналом Sh. Особливістю даної моделі є можливість переходу з режиму установки в режим роботи автомата у будь-який момент часу.

На рис. 3.9 наведено результат моделювання VHDL-моделі легкотестованого автомата Мура (FSM_Sh) в режимі обходу графа автомата по гамільтоновому циклу. При Sh = 0 (від 0 нс до 600 нс) автомат працює в стандартному режимі, при Sh = 1 (від 600 нс до 1100 нс) – відбувається обхід всіх станів циклічно (гамільтонів цикл).

На рис. 3.10 наведені результати автоматизованого синтезу легкотестованої VHDL-моделі автомата Мура з додатковим входом Sh з використанням інструментальних засобів пакету САПР: XILINX ISE. При синтезі використовувався модельний пристрій: плата Spartan 3E, мікросхема FPGA XC3S500E, Package FG 320.

```
Sreg0_NextState: process (State, x1, x2, x3)
begin
  case State is
    when a1 => NextState <=a2;
    when a2 => if Sh = '1' then NextState <= a3;
                elsif x1='1' then NextState <= a4;
                else NextState <= a3;
                end if;
    end case;
end process;
-- Input_signals
y1<='1' when State=a2 else '0';
y2<='1' when State=a3 else '0';
end;
```

Рисунок 3.8– Фрагмент архітектури VHDL-моделі легкотестованого автомата Мура FSM_Sh

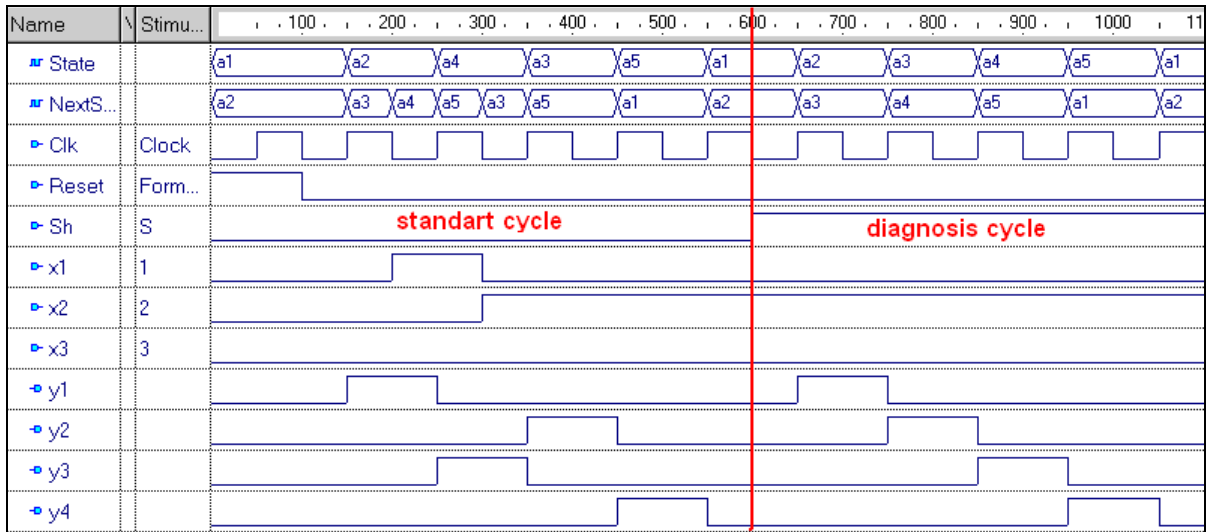


Рисунок 3.9 – Часова діаграма моделювання гамільтонового циклу автомата Мура з сигналом Sh (FSM_Sh)

На схемі автомату за результатами синтезу наступна відповідність сигналів (зліва позначення в VHDL-моделі, справа з САПР):

- Clk – Clk , Sh – In0, x1 – In1, x2 – In2, x3 – In3, Reset – Rst ,
- Q2 – FSM_FFd2 (внутрішній сигнал с вихода триггеру (Flip-Flop) D2)
- Q3 – FSM_FFd3 (внутрішній сигнал с вихода триггеру (Flip-Flop) D3)
- D2 – FSM_FFd2:In (внутрішній сигнал с вихода триггеру (Flip-Flop) D2)

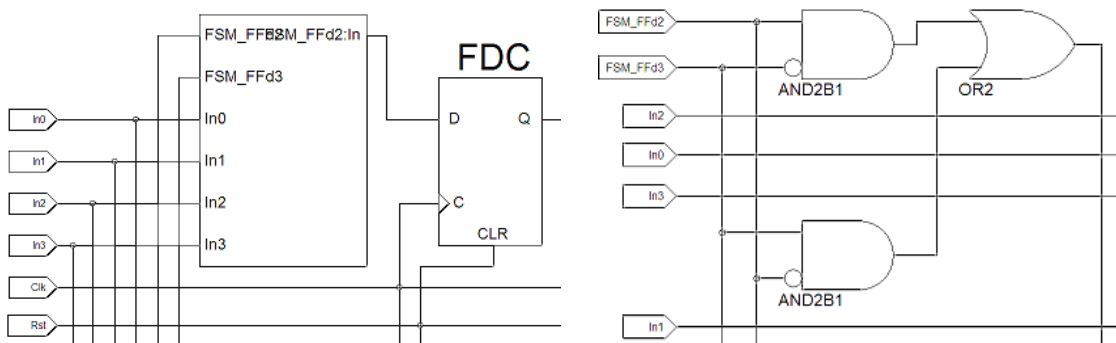


Рисунок 3.10 – Фрагмент схеми автомата Мура з сигналом Sh (FSM_Sh)

В аналітичному вигляді рівняння для функції збудження триггеру D2 буде:

$$D_2 = Q_2 Q_3 x_3 \overline{x_2} \overline{Sh} \vee \overline{Q_2} Q_3 x_1 \overline{Sh} \vee Q_2 Q_3 x_2 \overline{Sh} \vee \overline{Q_2} Q_3 \overline{x_1} \overline{Sh} \vee (Q_2 \overline{Q_3} \vee \overline{Q_2} Q_3) Sh$$

Аналіз рівняння показує наявність мультиплектора в синтезованій схемі, що підтверджує синтез легкотестованого автомату.

На рис. 3.11 представлений фрагмент архітектури VHDL-моделі легкотестованого автомата Мілі з додатковим сигналом Sh. Особливістю даної моделі є можливість переходу з режиму установки в режим роботи автомата у будь-який момент часу.

```
Sreg0_NextState: process (State, Sh, x1, x2, x3)
begin
  y1 <= '0'; y2 <= '0'; y3 <= '0'; y4 <= '0'; y5 <= '0';
  case State is
    when a1 => if Sh = '1' then NextState <= a2;
                else NextState <= a2; y1 <= '1'; y2 <= '1';
                end if;
    when a2 => if Sh = '1' then NextState <= a3;
                elsif x1='1' then NextState <= a4; y4<='1';
                else NextState <= a3; y3 <= '1';
                end if;
    when a3 => if Sh = '1' then NextState <= a4;
                else NextState <= a5;
                end if;
    when a4 => if Sh = '1' then NextState <= a5;
                elsif x2='1' then NextState<=a3; y4<='1'; y5<= '1';
                elsif x3='1' then NextState <= a4;
                else NextState <= a5; y3 <= '1';
                end if;
    when a5 => if Sh = '1' then NextState <= a1;
                else NextState <= a1; y5 <= '1';
                end if;
    when others => NextState <= a1;
  end case;
end process;
```

Рисунок 3.11 – Фрагмент архітектури VHDL-моделі легкотестованого автомата Мілі (FSM_Mealy_Sh)

На рис. 3.12 наведено результат моделювання VHDL-моделі легкотестованого автомата Мілі (FSM_Mealy_Sh) в режимі обходу графа автомата по гамільтоновому циклу. При Sh = 0 (від 0 нс до 400 нс) автомат працює в стандартному режимі, при Sh = 1 (від 400 нс до 950 нс) - відбувається обхід всіх станів циклічно (гамільтонів цикл).

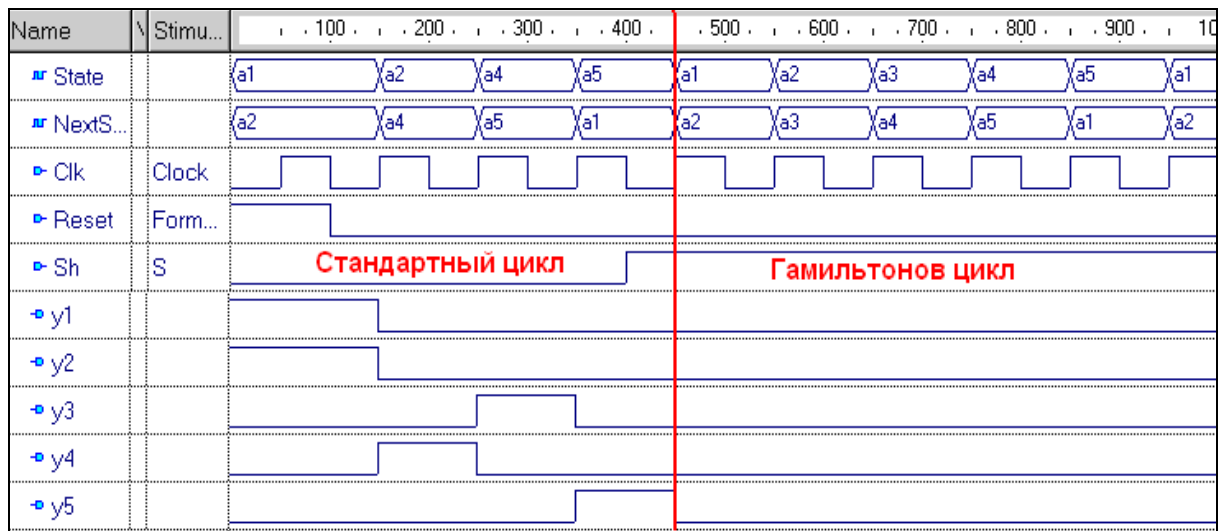


Рисунок 3.12 – Часова діаграма моделювання циклу Гамільтона автомата Мілі з сигналом Sh (FSM_Mealy_Sh)

Відзначимо одну особливість даної waveform, а саме, відсутність вихідних сигналів в режимі організації обходу графа переходів автомата (циклу Гамільтона). Це пов'язано з тим, що вихідні сигнали y_i автомата Мілі пов'язані з дугами графа переходів, а введення додаткових дуг Sh не передбачає наявності будь-якого вихідного сигналу. Звичайно, при побудові VHDL-моделі легкотестованого автомата кожній дузі Sh можна було б поставити у відповідність додатковий вихідний сигнал, але це привело б до розширення вихідного алфавіту та додатковим апаратурним витратам, що не відповідає постановці завдання.

На рис. 3.13 наведено результат синтезу VHDL-моделі з сигналом Sh (FSM_Mealy_Sh). Відповідність сигналів в системі синтезу XILINX ISE: Sh - In0, x1 - In1, x2 - In2, x3 - In3, Reset - Rst, Clk - Clk, Q2 - FSM_FFd2 (внутрішній сигнал з виходу тригера (Flip-Flop) D2),

Q3 - FSM_FFd3 (внутрішній сигнал з виходу тригера (Flip-Flop) D3), D2 - FSM_FFd2: In (внутрішній сигнал на вхід тригера (Flip-Flop) D2).

Повне рівняння функції збудження для тригера (розряду) D2 буде:

$$D_2 = Q_2 Q_3 x_3 \overline{x_2} \overline{Sh} \vee \overline{Q_2} Q_3 x_1 \overline{Sh} \vee Q_2 Q_3 x_2 \overline{Sh} \vee \overline{Q_2} Q_3 \overline{x_1} \overline{Sh} \vee (Q_2 \overline{Q_3} \vee \overline{Q_2} Q_3) Sh.$$

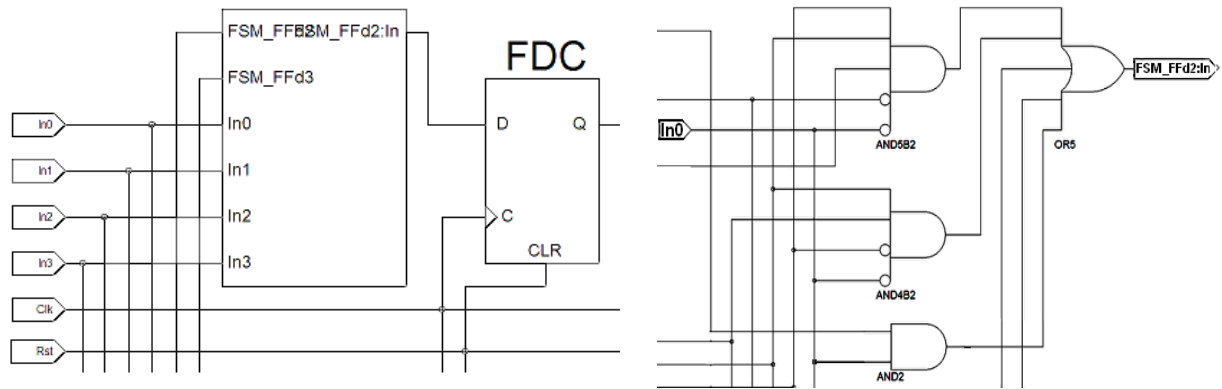


Рисунок 3.13 – Фрагмент синтезованої схеми автомата Мілі з сигналом Sh (FSM_Mealy_Sh)

Аналіз цього виразу показує, що відносно сигналу Sh фактично реалізується мультиплексор, що підтверджує синтез легкотестованої схеми з можливістю реалізації зсувного регістру у запам'ятовуючій частині автомата.

Результати схемної реалізації (синтезу) VHDL-моделей легкотестованих автоматів, реалізованих шляхом введення додаткових регістрів і мультиплексорів призводить до додаткових апаратних витрат, які можуть оцінюватися за Квайном. Витрати за Квайном визначаються як сумарне число входів всіх вентилів у схемі, оскільки число входів вентиля пропорційно числу транзисторів у ньому. Внутрішню структуру D триггера, з динамічним керуванням, можна потім додати в загальні затрати за Квайном.

Таблиця 3.1 – Апаратні витрати на схемну реалізацію легкотестованих VHDL-моделей КА

№	Тип автомату	VHDL-модель	Кількість тригерів	Витрати за Квайном
1	Мура	FSM	3	82
2	Мура	FSM_MX	3	131
4	Мура	FSM_Sh	3	104
5	Мілі	FSM_Mealy	3	110
6	Мілі	FSM_Mealy_Sh	3	130

Як видно з наведених результатів, оптимальним за апаратурними затратами є введення додаткового режиму підвищення керованості станів автомату (додатковий стовпчик Sh у ТПВ). Апаратурні затрати при тому збільшуються на 20-25% у залежності від типу автомату.

3.3 Аналіз апаратурних витрат при побудові легкотестованих автоматів

3.3.1 Аналіз тестопридатності функцій переходів

У підрозділі 3.2 з метою підвищення тестопридатності кінцевого керуючого автомату викладена концепція введення апаратурної надлишковості в модель абстрактного автомату, яка полягає у в доданні стовпця Sh у таблицю переходів-виходів (ТПВ) (додаткова дуга Sh у графі переходів автомата), що дає змогу встановити автомат в будь-який стан не більше, ніж за $n-1$ тактів. При автоматизованому синтезі вказаних автоматів з використанням моделей на мовах опису апаратури (HDL-моделей) введення додаткового стовпця реалізується шляхом додавання сигналу Sh в опис функції переходів у двохпроцесному автоматному шаблоні HDL-моделі. При синтезі це реалізувалося у додаткові мультиплексори, які дозволяли організувати сканування станів автомату в процесі проведення діагностичного експерименту. Зазначено, що додаткові апаратурні витрати при цьому не перевищують 25-30% в залежності від типу автомата.

Таким чином актуальною є задача мінімізації додаткових апаратурних витрат за рахунок організації оптимального обходу станів автомату. Метою даної роботи є аналіз апаратурних витрат при різних варіантах організації додаткового переходу між станами автомату в залежності від наявності безумовного переходу, умовного переходу та відсутності переходів між станами автомату, які аналізуються.

Розглянемо можливі варіанти додавання переходу з сигналом Sh між вершинами графу переходів (станів автомату) a_i та a_j .

Перший варіант полягає в додаванні Sh до безумовного переходу ($a_0 \rightarrow a_1$) у автоматі Мура. Відмітимо, що для використання наведених фрагментів графу переходів в процесі автоматизованого синтезу у графі мають враховуватися усі варіанти переходів () та зворотна дуга (що забезпечує відсутність вершин, які «висять»), рис. 3.14(а). ТПВ фрагментів автомату для початкового графу та графу з Sh наведені на рис. 3.14(б). В результаті між вершинами виявляються дві дуги з сигналами Sh та \overline{Sh} (рис. 3.14 в).

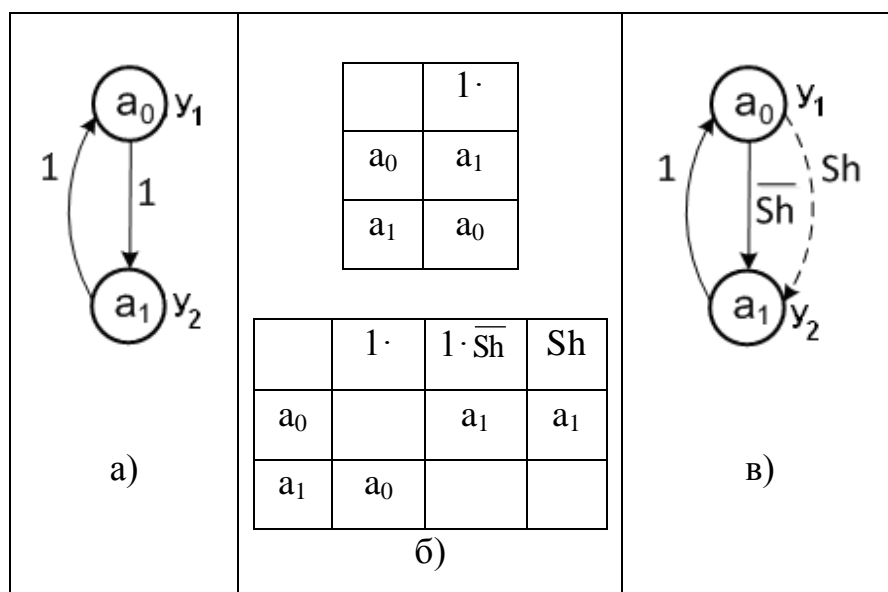


Рисунок 3.14 – Додавання Sh до безумовного переходу

В даному випадку функція переходу в стан a_1 матиме вигляд:
 $a_1 = a_0 Sh \vee a_0 \overline{Sh} = a_0$.

Застосовуючи закон склеювання, отримуємо a_0 , що означає відсутність додаткових витрат апаратури при реалізації мінімізованого виразу. Для кодування двох станів автомату використовується один D-тригер, a_0 кодується 0, a_1 кодується 1. Таким чином, вираз $a_1 = \overline{a_0}$ фактично реалізується інвертором.

Розглянемо VHDL-модель наведеного фрагменту автомату (рис. 3.15) та виконаємо її автоматизований синтез (рис.3.16).

```

architecture Moore of FSM is
    type State_type is (a0, a1);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset)
    begin
        if Reset='1' then State <= a0;
        elsif Clk'event and Clk = '0' then State <= NextState;
        end if;
    end process;

    Sreg0_NextState: process (State, Sh)
    begin
        case State is
            when a0=> if sh='1' then NextState <= a1;
                       else NextState <= a1;
                       end if;
            when a1=> NextState <= a0;
            when others => NextState <= a0;
        end case;
    end process;
    y1 <= '1' when State=a0 else '0';
    y2 <= '1' when State=a1 else '0';
end;

```

Рисунок 3.15 – VHDL-модель фрагменту автомату з безумовним переходом

Нижче наведені результати синтезу даної VHDL моделі з використанням пакету САПР: XILINX ISE, плата Spartan 3E, мікросхема FPGA XC3S500E, Package FG 320. На рис. 3.16 наведена схемна реалізація автомату, яка підтверджує попередні викладки.

Synthesizing Unit <FSM>.

Related source file is "C:/Temp/shkil2018/shkil_2states_2018/fsm.vhd".

Found 1-bit register for signal <State<0>>.

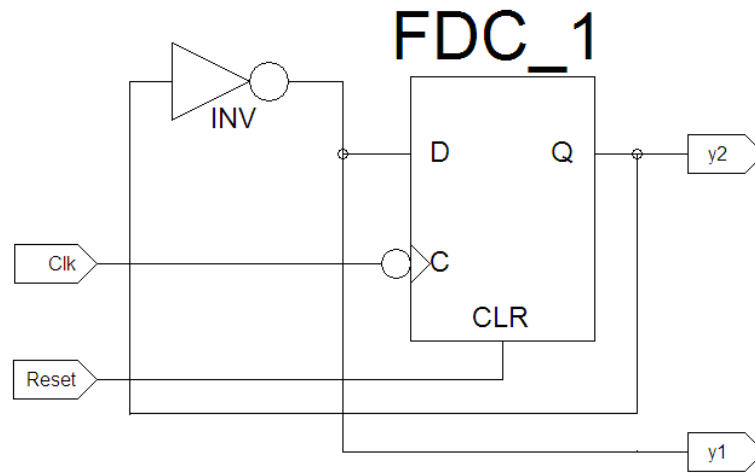


Рисунок 3.16 – Результати синтезу VHDL-моделі автомату з безумовним переходом

Другий варіант полягає в додаванні Sh до переходів між a_i і a_j при відсутності переходів в інші стани, відмінні від a_j . Ця ситуація характерна для автомата Мілі, але для аналізу цієї ситуації будемо використовувати змішаний автомат на три стани, у якого стани a_1 та a_2 відображаються моделлю Мура, а переходи ($a_0 \rightarrow a_1$) – моделлю Мілі. Граф переходів вказаного автомата зображений на рис. 3.17(а), а його ТПВ – на рис. 3.17(б). На рис.3.17(б) в ТПВ для спрощення сигнал \overline{Sh} не вказується. Граф переходів автомату з сигналами Sh та \overline{Sh} наведений на рис. 3.17(в).

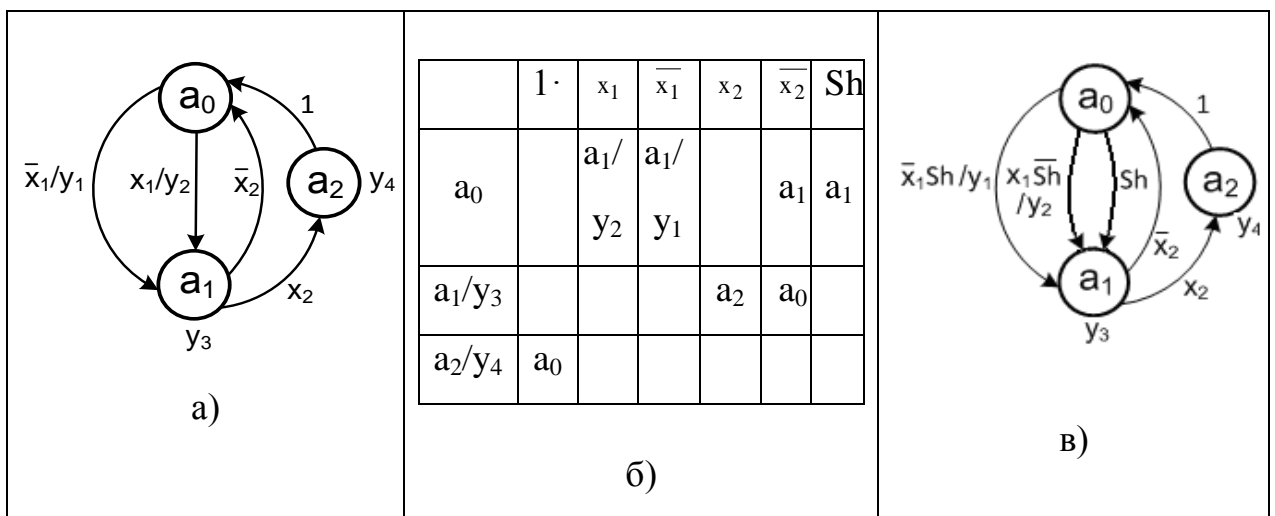


Рисунок 3.17 – Додавання Sh при наявності переходів тільки між a_0 та a_1

Функція переходу в стан a_1 для двох вихідних переходів і доданого переходу Sh буде мати вигляд:

$$a_1 = a_0 Sh \vee a_0 x_1 \overline{Sh} \vee a_0 \overline{x_1} \overline{Sh} = a_0 Sh \vee a_0 \overline{Sh} = a_0.$$

Застосовуючи поетапно закон склеювання, отримуємо a_0 , що означає відсутність додаткових витрат апаратури при реалізації мінімізованого виразу. Для будь-якого числа дуг, укладених між a_i і a_j , результат буде таким же. При реалізації вихідного виразу додаткові апаратурні витрати – додатковий вхід \overline{Sh} в усі вентиля, що реалізують терми вихідної функції переходів (рис. 3.18а) і вентиль, який реалізує терм $a_i Sh$ (рис. 3.18б).

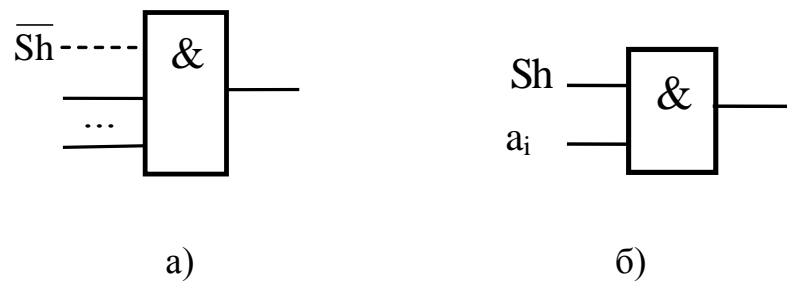


Рисунок 3.18 – Додаткові апаратурні витрати при наявності переходів тільки між a_0 та a_1

Розглянемо VHDL-модель наведеного фрагменту автомату (рис. 3.19) та виконаємо її автоматизований синтез (рис.3.20).

Found finite state machine <FSM_0> for signal <State>.

Analyzing FSM <FSM_0> for best encoding.

Optimizing FSM <State/FSM> on signal <State[1:2]> with gray encoding.

State | Encoding a0 | 00 a1 | 01 a2 | 11


```

library IEEE;
use IEEE.std_logic_1164.all;
entity FSM2 is
    port ( Sh, Reset,  x1, x2, Clk: in STD_LOGIC;
          y1, y2, y3, y4: out STD_LOGIC);
end;
architecture MooorMealy of FSM2 is
    type State_type is (a0, a1, a2);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset)
    begin
        if Reset='1' then State <= a0;
        elsif Clk'event and Clk = '0' then State <=
NextState;
        end if;
    end process;

    Sreg0_NextState: process (State, Sh, x1, x2)
    begin
        case State is
            when a0=>if sh='1' then NextState <= a1;
                elsif x1 = '1' then NextState <= a1; y2 <= '1';
                else NextState <= a1; y1 <= '1';
                end if;
            when a1=>if x2='1' then NextState <= a2;
                else NextState <= a0;
                end if;
            when a2=> NextState <= a0;
            when others => NextState <= a0;
        end case;
    end process;
    y3 <= '1' when State=a0 else '0';
    y4 <= '1' when State=a1 else '0';
end;

```

Рисунок 3.19 – VHDL-модель автомату з переходами тільки між a_0 та a_1

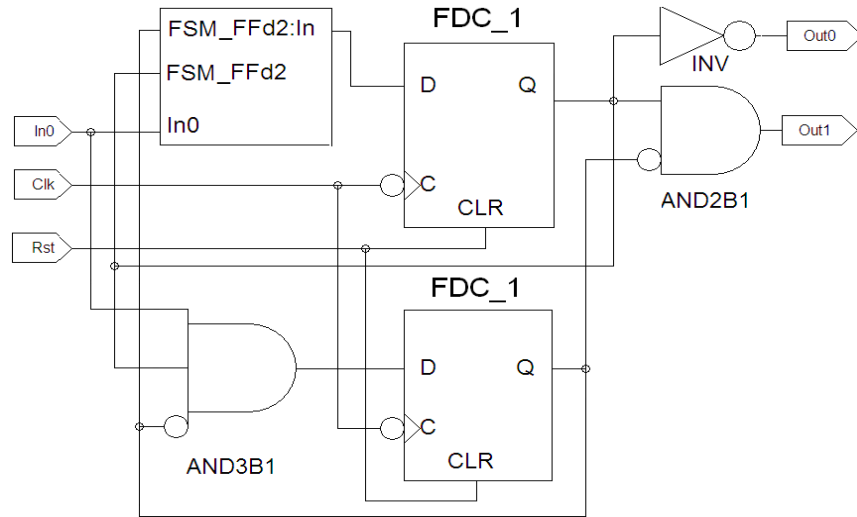


Рисунок 3.20– Результати синтезу VHDL-моделі автомату з переходами тільки між a_0 та a_1

Отримана схема підтверджує, що сигнал Sh (In0) через вентиль AND додається до функції збудження кожного з тригерів.

Третій варіант полягає в додаванні Sh до переходів між a_i і a_j при наявності переходів і в інші стани, відмінні від a_j (рис. 3.21а).

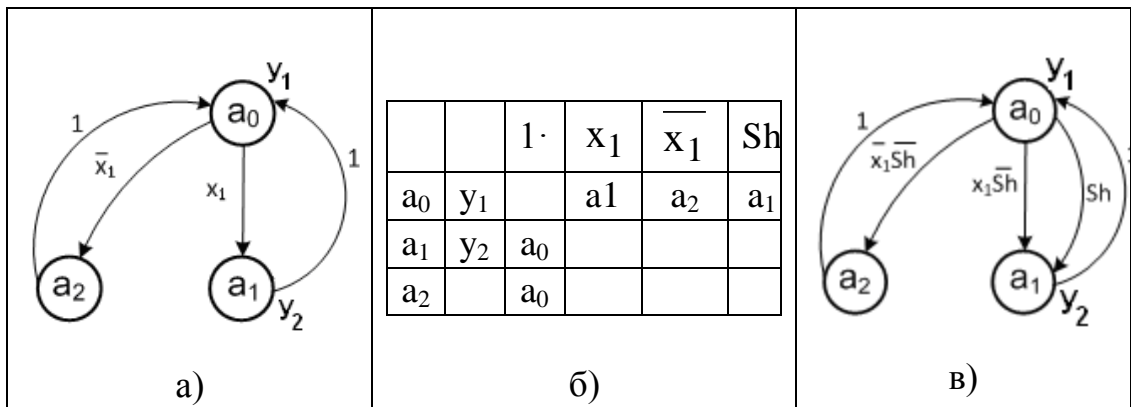


Рисунок 3.21 – Додавання Sh при наявності переходів між a_0 і a_1 та інших вихідних переходів

При цьому функція переходу в стан a_i (a_1) для одного вихідного переходу, доданого переходу Sh та переходу в стан $a_k \neq a_j$ (рис. 3.21в)

$$a_j = a_i \cdot Sh \vee a_i \cdot x_1 \cdot \overline{Sh} = a_i \cdot (Sh \vee x_1 \cdot \overline{Sh}) = a_i \cdot Sh \vee a_i \cdot x_1$$

Застосовуючи закон Блейка-Порецкого, отримуємо $a_i Sh \vee a_i x_1$, тобто додаткові апаратурні витрати при реалізації мінімізованого виразу для функції переходу в стан a_j – вентиль, який реалізує терм $a_i Sh$ (рис. 3.18б). При реалізації вихідного виразу для функції переходу в a_j додаткові апаратурні витрати - вентиль, який реалізує терм $a_i Sh$ плюс додатковий вхід в вентиль, який реалізує терм вихідної функції переходів (рис. 3.18а та 3.21в). Якщо переходів $a_i \rightarrow a_j$ в вихідному графі більше одного, то для кожного з цих переходів додається додатковий вхід в вентиль. Крім того, в даному варіанті, $\forall(a_i \rightarrow a_k), a_k \neq a_j$ переходи $a_i \rightarrow a_k$ також несуть додаткові апаратурні витрати у вигляді додаткового входу \overline{Sh} у вентиль, який реалізує терм вихідної функції переходів. Для прикладу на рис. 3.21(в) $a_k = a_i \overline{x_1} \overline{Sh}$.

```

architecture Moore of FSM3 is
    type State_type is (a0, a1, a2);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset)
    begin
        if Reset='1' then State <= a0;
        elsif Clk'event and Clk = '0' then State <=
NextState;
        end if;    end process;
    Sreg0_NextState: process (State, Sh, x)
    begin
        case State is
            when a0=> if sh='1' then NextState <= a1;
                elsif x = '1' then NextState <= a1;
                else NextState <= a2;
                end if;
            when a1=> NextState <= a0;
            when a2=> NextState <= a0;
            when others => NextState <= a0;
        end case;    end process;
    y1 <= '1' when State=a0 else '0';
    y2 <= '1' when State=a1 else '0';    end;

```

Рисунок 3.22 – VHDL-модель автомату при наявності переходів

$a_0 \rightarrow a_1$ та інших переходів

Розглянемо VHDL-модель наведеного фрагменту автомату (рис.3.22) та виконаємо її автоматизований синтез (рис.3.23).

Analyzing FSM <FSM_0> for best encoding.

Optimizing FSM <State/FSM> on signal <State[1:2]> with speed1 encoding.

State | Encoding a0 | 10 a1 | 01 a2 | 00

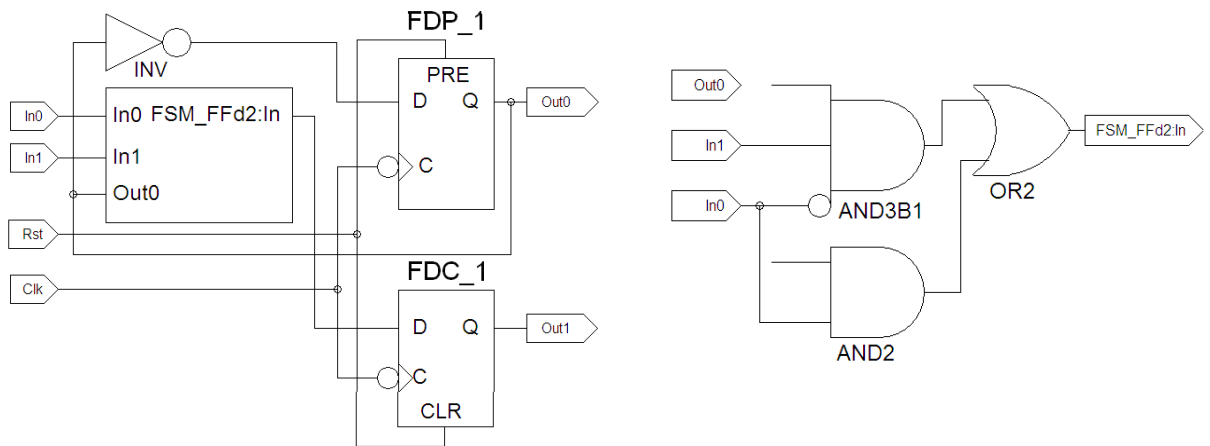


Рисунок 3.23 – Результати синтезу VHDL-моделі автомату наявності переходів між a₀ і a₁ та інших вихідних переходів

Четвертий варіант полягає в додаванні переходу з Sh між a_i і a_j при відсутності переходу a_i → a_j взагалі (рис. 3.24).

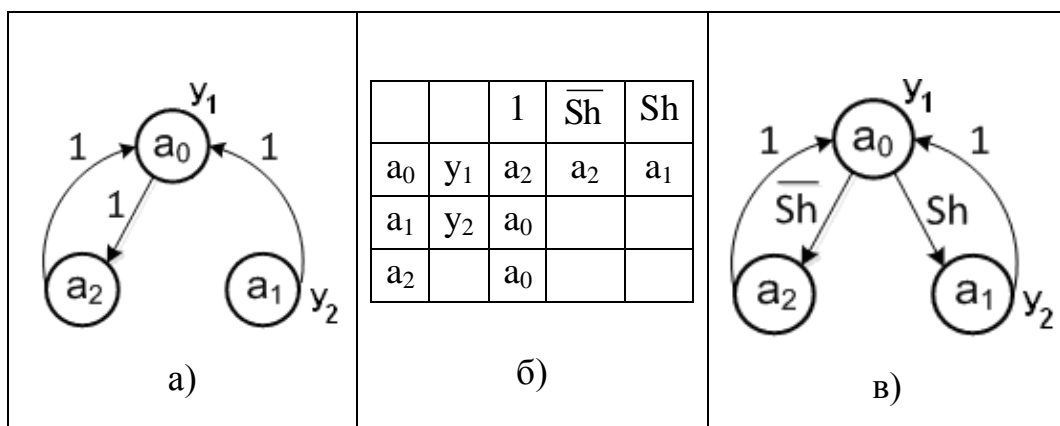


Рисунок 3.24 – Додавання Sh при відсутності переходів між a₀ і a₁

Тут додаткові апаратурні витрати – вентиль, який реалізує терм a_iSh

плюс додатковий вхід \overline{Sh} у вентилях, що реалізують $\forall(a_i \rightarrow a_k)$, $a_k \neq a_j$ терм вихідної функції переходів. Слід також зазначити, що $\forall(a_s \rightarrow a_j)$, $a_s \neq a_i$ на переходи $a_s \rightarrow a_j$ Sh не впливає.

Розглянемо VHDL-модель наведеного фрагменту автомату (рис.3.25) та виконаємо її автоматизований синтез (рис.3.26).

```

library IEEE;
use IEEE.std_logic_1164.all;
entity FSM4 is
    port ( Sh, Reset, Clk: in STD_LOGIC;
          y1, y2: out STD_LOGIC);
end;
architecture Moore of FSM4 is
    type State_type is (a0, a1, a2);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset)
    begin
        if Reset='1' then State <= a0;
        elsif Clk'event and Clk = '0' then State <=
NextState;
        end if;
    end process;
    Sreg0_NextState: process (State, Sh)
    begin
        case State is
            when a0=> if sh='1' then NextState <= a1;
                else NextState <= a2;
            end if;
            when a1=> NextState <= a0;
            when a2=> NextState <= a0;
            when others => NextState <= a0;
        end case;
    end process;
    y1 <= '1' when State=a0 else '0';
    y2 <= '1' when State=a1 else '0';
end;

```

Рисунок 3.25 – VHDL-модель автомату при відсутності переходів між a_0 і a_1

Optimizing FSM <State/FSM> on signal <State[1:2]> with speed1 encoding.

State | Encoding : a0 | 10 a1 | 01 a2 | 00

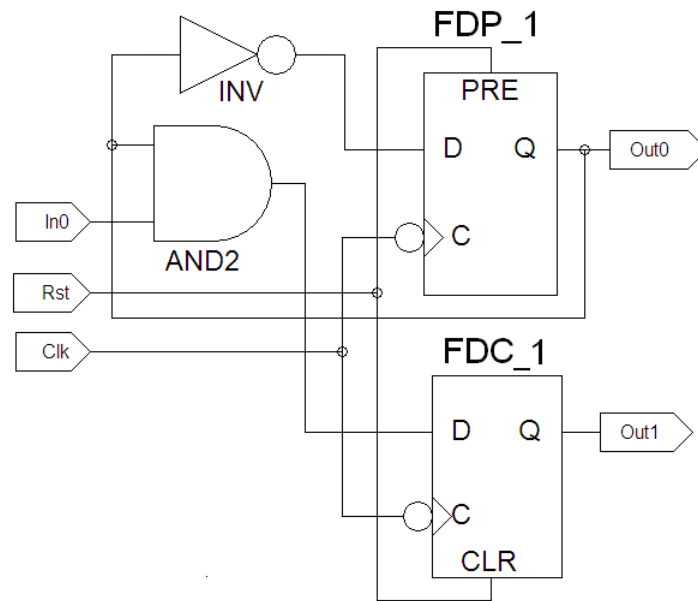


Рисунок 3.26 – Результати синтезу VHDL-моделі автомату відсутності переходів між a_0 і a_1

Підсумовуючи вищенаведене можна визначити наступне.

1. Для вибору шляху призначення додаткових переходів з Sh однозначно з точки зору мінімізації витрат апаратури по спадаючій пріоритету йде 1 варіант, за ним другий, а потім 3 або 4.

2. При обранні 1-го або 2-го варіанту при реалізації не мінімізованого виразу функції переходів, краще обирати 1-й, так він менше за кількістю вихідних дуг, а значить, буде менше додаткових входів \overline{Sh} в вентилі. При реалізації мінімізованого виразу функції переходів, немає різниці 1-й або 2-й, в обох випадках немає додаткових витрат апаратури.

3. Для обрання між 3 і 4 варіантами визначальним швидше буде число вихідних дуг, що задовольняють умові: $\forall(a_i \rightarrow a_k), a_k \neq a_j$. В якому варіанті їх менше, такий і має бути вибраним, оскільки додатковий вентиль на перехід $a_i \rightarrow a_j$ з Sh буде в обох випадках. Це буде суттєвим для способу реалізації

мінімізованого виразу функції переходів.

Для способу реалізації немінімізованого виразу функції переходів потрібно в варіанті 3 для підрахунку числа дуг для порівняння з варіантом 4 до умови: $\forall(a_i \rightarrow a_k)$, $a_k \neq a_j$, додати $\forall(a_i \rightarrow a_j)$ крім $a_i \rightarrow a_j$ з Sh.

Таким чином при обранні пари станів автомата, між якими встановлюється додатковий перехід (дуга Sh у графі переходів автомата), обирається той стан-наступник, для якого сумарна оцінка апаратурних витрат для функцій збудження мінімальна з урахуванням кодування станів автомата. При цьому відсутність переходів між станами прирівнюється до переходу з однією умовою.

3.3.2 Розрахунок тестопридатності графових моделей автоматів

Як було показано у підрозділі 3.3 (п. 3.3.1) додаткові апаратурні витрати, які забезпечують тестопридатність схемної реалізації керуючого автомата шляхом введення додаткових переходів, які дозволяють встановити автомат в довільний стан, залежить від складності функцій переходів та способу кодування станів автомата. Аналіз показав, що існує прямо-пропорційна залежність між складністю функцій переходів (функцій збудження), та надлишковими апаратурними витратами на реалізацію додаткової дуги Sh в графі переходів КА. Таким чином, існує можливість оптимізувати (зменшити) апаратурні витрати на забезпечення тестопридатності шляхом визначення оптимального порядку встановлення додаткових дуг Sh в графі переходів керуючого автомата.

Для кількісної оцінки апаратурних витрат на реалізацію функцій збудження, які забезпечують встановлення КА у довільний стан a_i , будемо використовувати показник досяжності.

В технічних системах управління під досяжністю розуміють можливість (складність за часом та діями) встановлення системи в зазначений технічний стан [91].

Стосовно КА в системах автоматного управління під досяжністю будемо розуміти ваговий коефіцієнт, що ставиться у відповідність кожній вершині графа переходів КА, який характеризує в умовних одиницях складність установки автомата у зазначений стан, починаючи з початкового стану.

При цьому слід враховувати наступне. Складність функцій виходів безпосередньо впливає на спостережуваність стану автомата та непрямим способом впливає на складність проведення діагностичного експерименту. Але для вирішення задачі встановлення КА у довільний стан складність функцій виходів можна не враховувати, тому що додавання додаткової дуги Sh впливає на складність апаратної реалізації функцій збудження, а функція виходів при цьому залишається незмінною.

Розрахунок досяжності КА будемо проводити шляхом аналізу змістовного графу переходів КА, в якому виконано умовне кодування станів автомата. В якості моделі КА будемо розглядати автомат Мілі, як найбільш близький к моделі структурного автомата.

Припустимо в графі переходів автомата Мілі існує вершина a_n , яка має k попередників (рис. 3.27) та відповідно k вхідних дуг.

Для кожної k -ї дуги графа переходів $(a_{m+k} \rightarrow a_n)$ обчислюється її коефіцієнт досяжності W_{m+k} , який на відміну від [92] враховує не тільки сумарну вагу умов переходу, а й співвідношення двійкових кодів станів автомата a_{m+k} та a_n .

$$W_{m+k} = q_{m+k} * \sum_{i=1}^R p_r, \quad (3.8)$$

де R – кількість умов переходу на k -й дузі, p_r – вага відповідної умови переходу, q_{m+k} – кодова відстань між двійковими кодами станів автомата a_{m+k} та a_n .

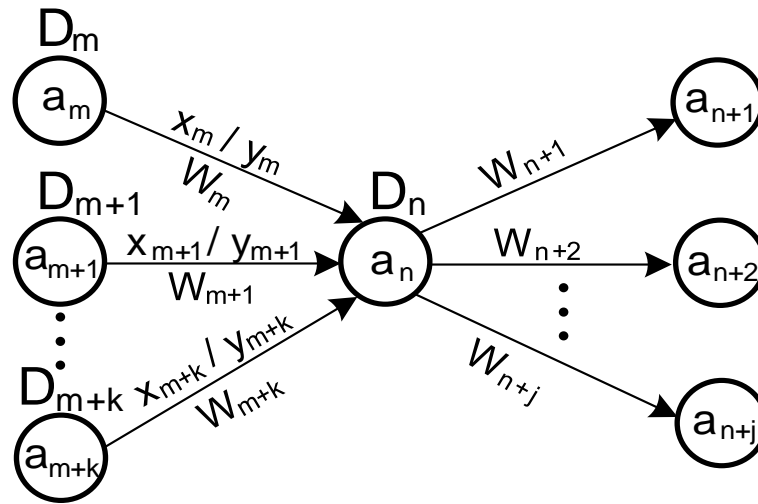


Рисунок 3.27 –Коефіцієнти досяжності в графі переходів КА

Зазначимо, що для мікропрограмних автоматів в обчислювальних пристроях $p_r = 1$. Для КА в системах логічного управління p_r обумовлюється складністю пристрою перетворення аналогових сигналів датчиків в логічні значення $x_r = \{0, 1\}$ в залежності від меж допустимих значень вхідних сигналів.

Досяжність вершини графа a_n обчислюється як мінімальна сума досяжностей вершин-попередників та коефіцієнтів досяжності W_{m+k} для дуг $(a_{m+k} \rightarrow a_n)$:

$$D_n = \min_{k=1}^K (D_{m+k} + W_{m+k}). \quad (3.9)$$

Відмітимо, що для початкової вершини a_0 $D_0 = 1$, петлі при розрахунку досяжності не враховуються.

Запропонуємо такий метод обчислення досяжностей в графі КА.

1. Початкова вершина графа переходів КА a_0 помічається і їй присвоюється початкова досяжність $D_0 = 1$.

2. Для кожної поміченої вершини обчислюються коефіцієнти досяжності всіх вихідних дуг W_j , де $j = \overline{1, J}$ – кількість вихідних дуг, за формулою (3.8) та ці дуги помічаються.

3. Для всіх вершин, для яких помічені дуги є вхідними обчислюється D_n за формулою (3.9). Зазначені вершини помічаються.

4. Пункти 2 та 3 повторюються доти, поки усі вершини графа не будуть помічені.

Відмітимо, що при обранні пари станів автомата, між якими встановлюється додатковий перехід (дуга Sh у графі переходів автомата), обирається той стан-наступник, для якого сумарна оцінка апаратурних витрат для функцій збудження мінімальна з урахуванням кодування станів автомата. При цьому відсутність переходів між станами прирівнюється до переходу з однією умовою.

При обранні переходу $(a_n \rightarrow a_j)$, на якому розміщується додаткова дуга Sh , у разі наявності двох та більше вихідних дуг з вершини a_n ,

використовується функція переваги $F = \min_{j=1}^J |D_n - D_j|$, де D_n – досяжність

вершини, з якої виходить додаткова дуга Sh , D_j – досяжність вершини-наступника, $j = \overline{1, J}$ – кількість вихідних дуг з вершини D_n . Виходячи з цього додаткова дуга ставиться на переході, для якого функція приналежності мінімальна.

Наведений метод обчислення досяжностей та розстановки додаткових дуг Sh є евристичним з двох причин. По-перше, на обчислення досяжності впливає спосіб кодування станів автомата, а при автоматизованому синтезі проектувальник не завжди впливає на спосіб кодування станів, який забезпечує мінімальні апаратурні витрати. По-друге, вибір на кожному кроці алгоритму тільки однієї дуги, яка визначає досяжність наступної вершини, фактично обумовлює ціну тільки одного зі шляхів у графі, за яким автомат

може бути встановлений у зазначений стан. А це не враховує циклів у графі переходів автомата, що в кінцевому рахунку може вплинути на обчислення досяжності.

Одним з результатів використання коефіцієнтів досяжності є обчислення тестопридатності по забезпеченню діагностичного експерименту по обходу графа переходів за визначеним p -м шляхом. Показник складності p -го шляху S_p обчислюється за формулою (3.10):

$$S_p = \sum_{i=1}^H W_{m,n} , \quad (3.10)$$

де H – кількість дуг для p -го шляху. В якості переваги при обранні шляху обходу графа переходів обирається той, у якого S_p мінімальна.

3.4 Процедура розміщення додаткових дугу у графі переходів

В якості прикладу побудови тестопридатного керуючого автомату (КА) розглянемо мікропрограмний автомат (МПА) складання чотирирозрядних двійкових знакових чисел у додатковому модифікованому коді. Для цієї мікропрограми в якості КА розглядається автомат Мура [74]. На рис. 3.28 наведені граф-схема алгоритму функціонування пристрою складання та змістовний орієнтований граф переходів його КА за моделлю Мура. Кодування станів автомата здійснюється з використанням системи автоматизованого синтезу на основі моделі на мовах опису апаратури.

Як було показано у підрозділі 2.2 неруйнівний діагностичний експеримент передбачає обхід всіх вершин графа, починаючи з початкової та з поверненням у початковий стан, тобто гамільтонів цикл у графі переходів.

Гамільтонів цикл – це замкнутий маршрут, що включає кожну вершину графа тільки один раз, при чому не обов'язково всі дуги (ребра) мають

входити в обхід. Граф називається гамільтоновим, якщо він містить гамільтонів цикл. За визначенням існування гамільтонових циклів в довільному орієнтованому графі переходів не гарантовано. Довільний граф може мати один гамільтонів цикл, декілька гамільтонових циклів, або не мати їх зовсім. Для реалізації неруйнівного ДЕ по обходу всіх вершин графа, він має бути перетворений у гамільтонів за рахунок введення додаткових дуг Sh.

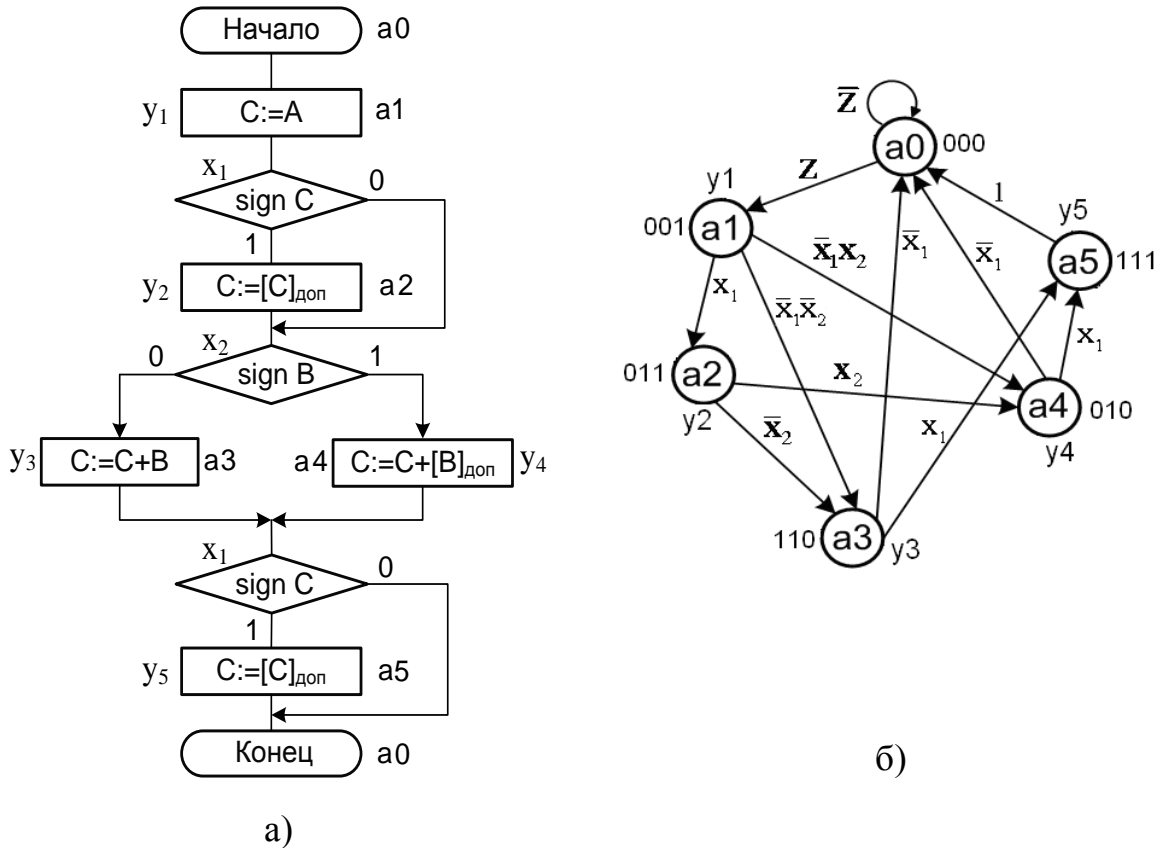


Рисунок 3.28 – Фрагмент ГСА МПА складання (а) та граф переходів КА (б)

Оптимальним з точки зору часу проведення буде такий ДЕ, в якому буде мінімальна кількість гамільтонових циклів. Достатньою умовою існування гамільтонова циклу в графі є умови Дірака та Оре.

Нехай $|V|=p$ – число вершин у графі $G = \langle V, U \rangle$; якщо ступінь (degree) кожної вершини не менше, ніж $p/2$, тобто $\forall v_i \in V \text{ deg } v_i \geq \lfloor |V|/2 \rfloor$,

де $[]$ – найближче ціле число, то граф називається графом Дірака, який в свою чергу є гамільтоновим.

При наявності декількох гамільтонових циклів в графі застосовується умова Оре. Нехай p – кількість вершин в даному графі і $p > 2$. Якщо для будь-якої пари несуміжних вершин (x, y) виконано нерівність $\deg x + \deg y > p$, то даний граф – гамільтонів (іншими словами: сума ступенів будь-яких двох несуміжних вершин не менше загального числа вершин в графі).

Теорема Бонді-Хватала узагальнює умови Дірака і Оре. Граф є гамільтоновим тоді і тільки тоді, коли його замикання – гамільтонів граф. Для графа G з n вершинами замикання будується додаванням в G ребра (u, v) для кожної пари несуміжних вершин u і v , сума ступенів яких не менше n [114].

Перевіримо виконання умови Дірака. Для графа на рис. 3.28 (б) ступені вершин наведені на рис. 3.29 (а). Для цього графа умова Дірака виконується: $p = |V| = 6$, $p/2 = |V|/2 = 3$, $\deg(a_0) = 6 > 3$, $\deg(a_1) = 4 > 3$, $\deg(a_2) = 3$, $\deg(a_3) = 6 > 3$, $\deg(a_4) = 6 > 3$, $\deg(a_5) = 3$. Таким чином, граф є графом Дірака, тобто гамільтоновим.

Таким чином, на основі теореми Бонді-Хватала для графа МПА для побудови множини гамільтонових циклів можна додати три пари несуміжних вершин, $(a_1 \leftrightarrow a_5)$, $(a_2 \leftrightarrow a_5)$ $(a_3 \leftrightarrow a_4)$. Проаналізувавши запропоновані додаткові дуги в графі можна зробити висновок, що тільки додатковий перехід $(a_3 \leftrightarrow a_4)$ дозволяє організувати єдиний гамільтонів цикл в цьому графі. На рис. 3.29 (б) наведений модифікований граф переходів, який має одне гамільтонове замикання.

Для мінімізації кількості гамільтонових циклів в графі шляхом введення додаткових дуг Sh між тими вершинами, де в початковому графі переходів не було, будемо використовувати умови теореми Бонді-Хватала.

З точки зору мінімізації апаратних витрат перевага віддається такому шляху обходу графа КА, для якого функція переваги (вага p -го шляху) буде

$S_P = \min \sum_{i=1}^H W_{m,n}$. При цьому слід враховувати, що вага додаткових дуг в графі для неіснуючих переходів дорівнює 1.

Для графа переходів КА на рис.3.29 (б) обчислюємо вагу всіх дуг (коефіцієнти досяжності W_{m+k}) та сформуємо матрицю суміжності з розрахунком досяжностей вершин у останньому рядку матриці, яка наведена на рис. 3.30.

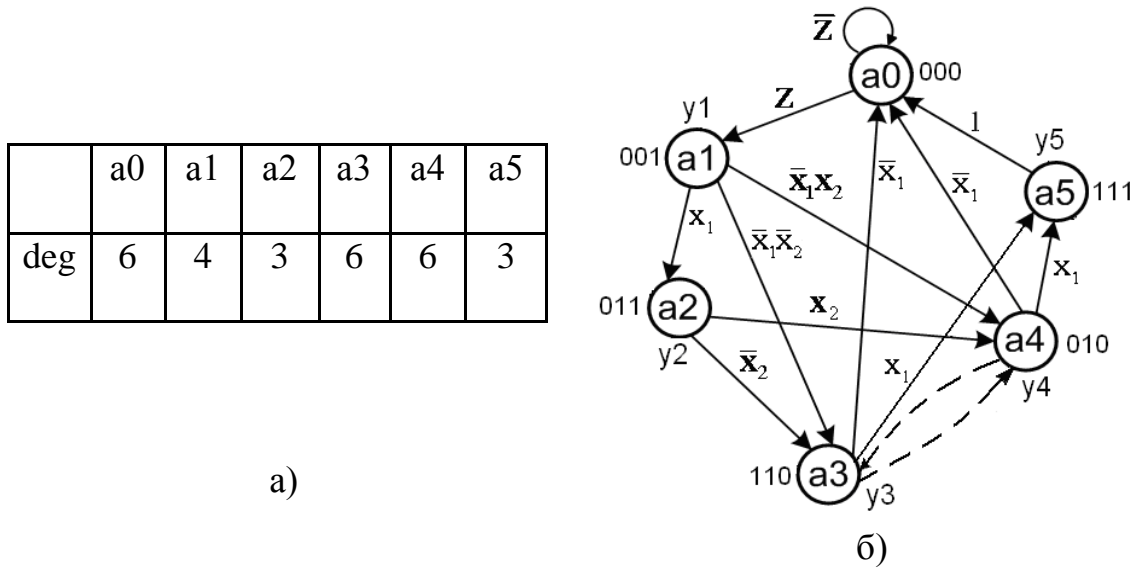


Рисунок 3.29 – Умови Дірака (а) та варіанти розміщення додаткових дуг у графі переходів його КА (б)

	a0	a1	a2	a3	a4	a5	a0
a0		1					
a1			1	6	4		
a2				3	1		
a3	2				1	1	2
a4	1			1		2	1
a5	3						3
D	1	2	3	6	4	6	9

Рисунок 3.30 – Матриця суміжності для графа МПА з розрахунком досяжностей

Сума коефіцієнтів досяжності по обходу вершин графа переходів для шляху $a_0-a_1-a_2-a_3-a_4-a_5-a_0$ буде $S_1 = 1+1+4+1+2+3 = 13$, а для шляху $a_0-a_1-a_2-a_4-a_3-a_5-a_0$ буде $S_2 = 1+1+1+1+1+3 = 8$. Тобто другий шлях має меншу вагу та можна припустити, що додаткові дуги S_h за цим шляхом обходу графа дадуть менші надлишкові апаратні витрати. Це припущення буде перевірено у розділі 4 при автоматизованому синтезу схемних реалізацій КА з різними додатковими дугами.

Серед математичних способів визначення гамільтонових циклів у графі можна розглядати методи перебору, алгебраїчний метод, а також методи гілок і границь та динамічного програмування. Останні два застосовуються для пошуку гамільтонових циклів на зважених графах. Для вирішення задачі пошуку мінімального за вагою гамільтонова циклу будемо використовувати метод гілок та границь.

Для графа на рис. 3.29 (б) сформуємо матрицю суміжності розміру $[6 \times 6]$, яка задає вартість всіх попарних зв'язків між вершинами, причому відповідні елементи приймаються $c_{ij} = \infty$, якщо відсутні дуги між вершинами i, j . (рис. 3.31). В якості ваги переходу будемо використовувати коефіцієнти досяжності. Наведемо покрокове виконання методу гілок та границь для цього графу.

i/j	a0	a1	a2	a3	a4	a5
a0	∞	1	∞	∞	∞	∞
a1	∞	∞	1	6	4	∞
a2	∞	∞	∞	3	1	∞
a3	2	∞	∞	∞	1	1
a4	1	∞	∞	1	∞	2
a5	3	∞	∞	∞	∞	∞

Рисунок 3.31 – Матриця суміжності $[6 \times 6]$

1. З кожного рядка матриці вартості віднімаємо мінімальний елемент. Результуюча матриця наведена на рис.3.32.

i/j	a0	a1	a2	a3	a4	a5	
a0	∞	0	∞	∞	∞	∞	1
a1	∞	∞	0	5	3	∞	1
a2	∞	∞	∞	2	0	∞	1
a3	1	∞	∞	∞	0	0	1
a4	0	∞	∞	0	∞	2	1
a5	0	∞	∞	∞	∞		3

Рисунок 3.32 – Матриця суміжності [6 x 6] з відніманням

2. У кожному стовпчику є нуль, тому етап віднімання мінімальних елементів зі стовпчиків відсутній.

3. Підраховуємо сумарний від'ємник і приймаємо це значення як нижню границю (Н.Г.) вартостей всіх розв'язків: Н.Г.=1+1+1+1+1+3=8.

4. Розміщення міток над нулями. Розглядаються нульові елементи матриці. Над кожним нулем вказується числова мітка, обумовлена так: у припущенні, якщо розглянутий нуль замінити на безкінечність, то з рядка й стовпця, на перетинанні яких він знаходиться, можна відняти мінімальні елементи, які в сумі й визначають мітку. Для визначення мітки можна міняти на безкінечність тільки один розглянутий у цей момент нуль. Якщо нуль не єдиний у рядку або стовпці, то мінімальний елемент матриці неможна відняти з рядка чи стовпця, оскільки негативні значення для вартостей не допускаються.

Так, над нульовим елементом, розташованим на позиції c_{01} , зазначена мітка 0, оскільки не існує жодного ненульового елемента у рядку $i=0$ та стовпчику $j=1$, на перетині яких він знаходиться. Над нульовим елементом,

розташованим на перетині рядка $i=1$ та стовпця $j=2$, ставиться мітка 3, оскільки з рядка можна винести (вирахувати) 3. Над нульовим елементом, розташованим на перетині рядка $i=4$ та стовпця $j=3$, ставиться мітка 2, оскільки з рядка можна вирахувати число 2. Аналогічно розставляються числові мітки над іншими нулями матриці (рис. 3.33).

i/j	a0	a1	a2	a3	a4	a5
a0	∞	0^0	∞	∞	∞	∞
a1	∞	∞	0^3	5	3	∞
a2	∞	∞	∞	2	0^2	∞
a3	1	∞	∞	∞	0^0	0^1
a4	0^0	∞	∞	0^2	∞	1
a5	0^0	∞	∞	∞	∞	∞

Рисунок 3.33 – Матриця суміжності [6 x 6] з мітками

Вибирається той нуль, що при заміні на безкінечність дозволяє відняти найбільшу сумарну кількість із рядка й стовпця, на перетинанні яких він розташований, тобто нуль з максимальною міткою. У даній матриці такий нуль має мітку 3 і відповідає переходу з вершини 1 до вершини 2: $1 \rightarrow 2$. Множина всіх розв'язків розщеплюється на дві групи маршрутів, де переїзд із 1 в 2 дозволений і де такий переїзд не допускається: $1 \rightarrow 2$ і $1 \not\rightarrow 2$. Дерево розв'язків розбивається на два піддерева, причому групи маршрутів із припустимими переїздами прийнято розташовувати в лівих вузлах – піддеревах (рис. 3.34). Нижня границя маршрутів у правому піддереві із забороненими переїздами збільшується на величину, рівну максимальній мітці над нулем 3: Н.Г. ($1 \not\rightarrow 2$) = $8 + 3 = 11$.

Після розщеплення всієї групи маршрутів на зазначені підгрупи слід понизити порядок матриці шляхом викреслювання рядка й стовпця, на

перетині яких розташований нуль із максимальною міткою – це рядок з номером $i=1$ і стовпець із номером $j=2$ (рис. 3.35 а)

Оскільки виконано перехід з вершини a_1 у вершину a_2 і зворотне повернення неможливо (кожну вершину можна відвідувати тільки один раз, за винятком першої та останньої), слід заборонити зворотний перехід з a_2 в a_1 , припускаючи в новій матриці $c_{21} = \infty$. Проте такий елемент вже є у модифікованій матриці після зниження її порядку. У перетвореній матриці зберігається нумерація рядків і стовпців (рис. 3.35 б).



Рисунок. 3.34 – Розщеплення всіх розв'язків на дві підгрупи

i/j	a0	a1	a2	a3	a4	a5
a0	∞	0^{10}	∞	∞	∞	∞
a1	∞	∞	0^{13}	5	3	∞
a2	∞	∞	∞	2	0^{12}	∞
a3	1	∞	∞	∞	0^{10}	0^{11}
a4	0^{10}	∞	∞	0^{12}	∞	1
a5	0^{10}	∞	∞	∞	∞	∞

а)

i/j	a0	a1	a3	a4	a5
a0	∞	0	∞	∞	∞
a2	∞	∞	2	0	∞
a3	1	∞	∞	0	0
a4	0	∞	0	∞	1
a5	0	∞	∞	∞	∞

б)

Рисунок 3.35 – Зниження порядку матриці суміжності

Після зниження порядку матриці виконується оцінка нижньої границі вартостей маршрутів у лівій частині піддерева. Для цього з'ясовується, чи є

нули в кожному рядку/стовпці перетвореної матриці. Видно, що у кожному рядку та кожному стовпчику є нуль. Тому нижня границя групи маршрутів зліва не змінюється: Н.Г. $(1 \rightarrow 2) = 8$.

5. Процедура розставлення міток над нулями й зниження порядку застосовується до перетвореної матриці розміру 5×5 (рис. 3.36).

i/j	a0	a1	a3	a4	a5
a0	∞	0^{10}	∞	∞	∞
a2	∞	∞	2	0^{12}	∞
a3	1	∞	∞	0^{10}	0^{11}
a4	0^{10}	∞	0^{12}	∞	1
a5	0^{10}	∞	∞	∞	∞

Рисунок 3.36 – Матриця суміжності $[5 \times 5]$

Знову вибирається нуль, що при заміні на безкінечність дозволяє відняти найбільшу сумарну кількість зі свого рядка й свого стовпця (див. п. 4). Цей нуль відповідає переходу з вершини 4 до вершини 3: $4 \rightarrow 3$. Множина всіх розв'язків розщеплюється на дві групи маршрутів: $4 \rightarrow 3$ і $4 \not\rightarrow 3$ (рис. 3.37). Нижня границя маршрутів у правому піддереві при цьому збільшується на 2 – максимальну мітку над нулем: Н.Г. $(4 \not\rightarrow 3) = 8 + 2 = 10$.

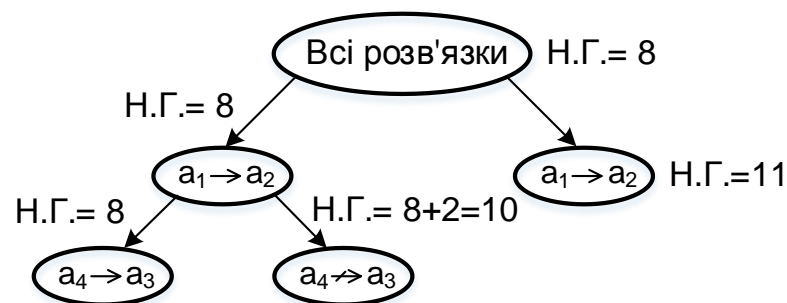


Рисунок 3.37 – Розщеплення групи розв'язків у лівому вузлі $1 \rightarrow 2$ на дві підгрупи

Зниження порядку матриці виконується шляхом видалення з розгляду рядка з номером $i=4$ і стовпця з номером $j=3$, після чого в новій матриці приймається $c_{30} = \infty$, (забороняється перехід з 3 в 0, оскільки цикл замкнеться передчасно, коли не всі вершини були включені до маршруту). У перетвореній матриці зберігається нумерація рядків і стовпців (рис.3.38).

i/j	a0	a1	a3	a4	a5
a0	∞	0^{10}	∞	∞	∞
a2	∞	∞	2	0^{12}	∞
a3	1	∞	∞	0^{10}	0^{11}
a4	0^{10}	∞	0^{12}	∞	1
a5	0^{10}	∞	∞	∞	∞

а)

i/j	a0	a1	a4	a5
a0	∞	0^{10}	∞	∞
a2	∞	∞	0^{12}	∞
a3	∞	∞	0^{10}	0^{11}
a5	0^{10}	∞	∞	∞

б)

Рисунок 3.38 – Зниження порядку матриці суміжності

б. Внаслідок перетворень отримано матрицю, що містить тільки безкінечності та нулі. За побудованим деревом розв'язків отримано фрагменти гамільтонова циклу, а саме: $a1 \rightarrow a2$, $a4 \rightarrow a3$. Вони складають гамільтонів цикл, якщо їх пов'язати між собою за допомогою переходів, що відбудовуються по нульовим значенням матриці: $a0 \rightarrow a1$, $a2 \rightarrow a4$, $a3 \rightarrow a5$, $a5 \rightarrow a0$. Отже, отримано гамільтонів цикл мінімальної довжини 8, тобто шлях $a0 \rightarrow a1 \rightarrow a2 \rightarrow a4 \rightarrow a3 \rightarrow a5 \rightarrow a0$.

Таким чином, математично підтвержені результати евристичного методу обрання оптимального гамільтонового шляху в графі переходів КА на основі використання формули (3.10).

3.5 Висновки до розділу 3

1. Запропоновано і математично обґрунтовано розширення вхідного алфавіту керуючого автомата за рахунок введення додаткового стовпця у таблицю переходів-виходів, що дозволяє встановлювати автомат в довільній стан за $(n-1)$ тактів, де n – кількість станів автомата.

2. Розроблені та верифіковані HDL-моделі легкотестованих автоматів Мілі та Мура за рахунок введення у HDL-код додаткової змінної (зовнішнього входу) Sh , що відповідає розширенню таблиці переходів-виходів. Верифікація проводилася з використанням системи моделювання Active-HDL. Аналіз апаратурних витрат показав, що оптимальним за апаратурними затратами є введення додаткового режиму підвищення керованості станів автомату (додатковий стовпчик Sh у ТПВ). Апаратурні затрати при тому збільшуються на 20-25% у залежності від типу автомату.

3. Проаналізовані апаратурні витрати на введення додаткового стовпця Sh в HDL- моделі керуючих автоматів для різних варіантів функцій переходів між станами автомата. Показано, що додаткові апаратурні витрати прямо пропорційні складності функцій переходів з урахуванням способу кодування станів автомату. Апаратурні витрати оцінювалися шляхом аналізу результатів автоматизованого синтезу HDL- моделей з використанням САПР XILINX ISE.

4. Запропоновані процедури оцінки тестопридатності графових моделей керуючих автоматів шляхом розрахунку досяжності вершин (станів) графу переходів. Це дозволило оптимальним (з точки зору додаткових апаратурних витрат) чином, розмістити додаткові дуги переходів Sh між відповідними станами автомата.

5. Розроблений евристичний метод побудови неруйнівного діагностичного експерименту на основі гамільтових циклів. Працездатність запропонованого методу підтверджено перевіркою результатів за допомогою математичного методу шляхів та гілок.

РОЗДІЛ 4

АВТОМАТИЗАЦІЯ ПРОЕКТУВАННЯ ТЕСТОПРИДАТНИХ АВТОМАТІВ

У четвертому розділі проведено апробацію й оцінку розроблених методів тестопридатного проектування HDL-моделей керуючих автоматів засобами САПР ПЛІС, наведено опис програмних продуктів, що забезпечують автоматизацію проектування тестопридатних КА, а також оцінено додаткові апаратурні витрати.

В ході проведення досліджень мають бути вирішені такі завдання :

1) проаналізувати апаратурні витрати при автоматизованому синтезі VHDL-моделей легкотестованих КА для різних варіантів розміщення додаткових дуг Sh;

2) розробити програмний модуль з візуальним інтерфейсом для вводу графової моделі КА та побудови легкотестованої VHDL-моделі;

3) виконати автоматизований синтез керуючого автомата КП ГРП з використання розроблених програмних засобів та оцінити надлишкові апаратурні витрати на забезпечення тестопридатності.

При автоматизованому синтезі використовувати інструментальні засоби САПР XILINX ISE на технологічній платформі ПЛІС Spartan 3E.

Результати досліджень опубліковані в роботах [118, 119, 122, 124].

4.1 Тестопридатне проектування керуючого автомату

В якості прикладу побудови тестопридатного керуючого автомату (КА) розглядатимемо мікропрограмний автомат (МПА) складання чотирирозрядних двійкових знакових чисел в додатковому модифікованому коді. Для цієї мікропрограми в якості КА розглядається автомат Мура. На рис 4.1(а) показаний фрагмент змістовної граф-схеми алгоритму вказаної мікропрограми з відмітками станів КА для автомата Мура. Кожній операторній вершині ГСА (стану автомата Мура) відповідає набір сигналів u_i ,

що керують, кожен з яких ініціює виконання певної мікрооперації операційного автомату. Для спрощення подальшого викладу в цьому фрагменті мікропрограми не аналізується результат складання на переповнювання розрядної сітки, що в цілому не міняє структури мікропрограми. На рис. 4.1(б) наведений змістовний граф переходів МПА Мура з урахуванням побудови гамільтонових циклів.

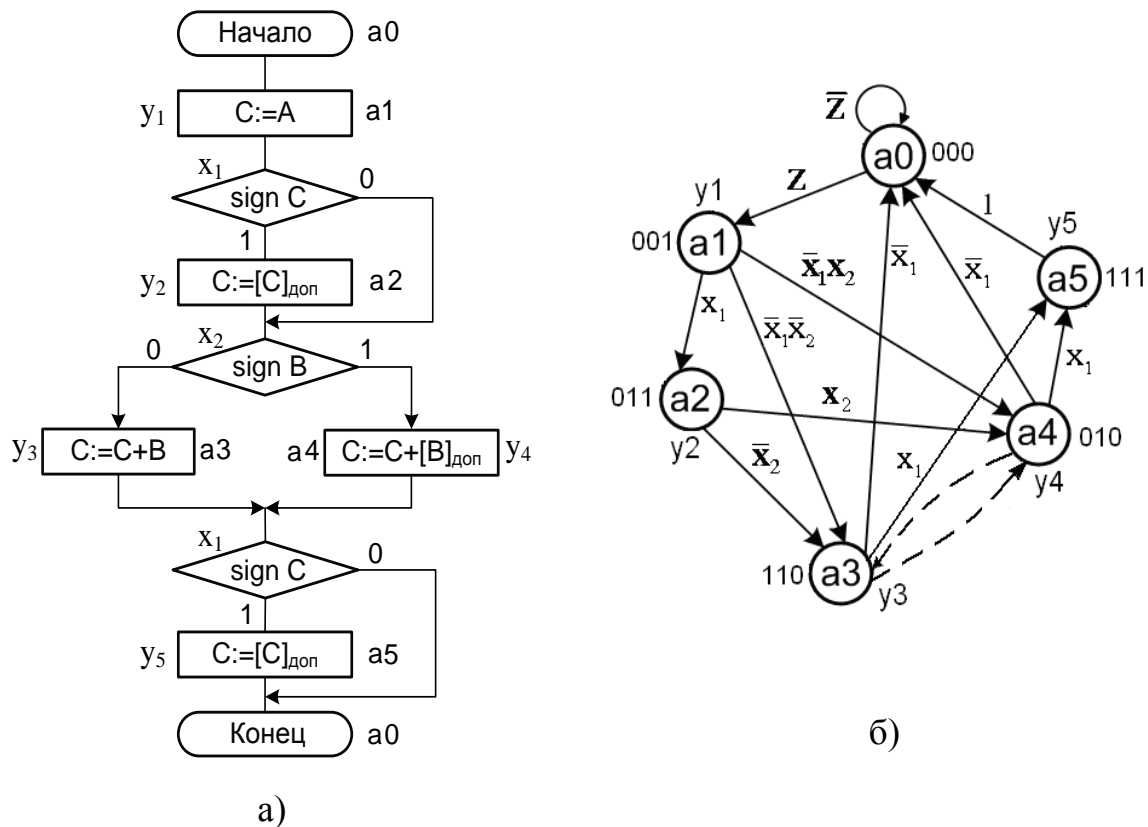


Рисунок 4.1 – Фрагмент ГСА МПА складання (а) та граф переходів КА (б)

На основі графа переходів побудована VHDL-модель керуючого автомата у формі двохпроцесного автоматного шаблону, яка представлена на рис. 4.2.

В підрозділі 3.3 зроблено припущення, що розміщення додаткових дуг Sh в графі переходів КА таким чином, якщо вони створюють гамільтонів цикл з мінімальною вагою, то додаткові апаратурні витрати мають бути мінімальними. Підтвердимо це результатами автоматизованого синтезу.

```

library IEEE;
use IEEE.std_logic_1164.all;
entity FSM is
    port (Reset, Z, Clk, x1, x2: in STD_LOGIC;
          y1, y2, y3, y4, y5: out STD_LOGIC);
end;
architecture Moore of FSM is
    type State_type is (a0, a1, a2, a3, a4, a5);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset)
    begin
        if Reset='1' then State <= a0;
        elsif Clk'event and Clk = '0' then State <=
NextState;
        end if;
    end process;
    Sreg0_NextState: process (State, x1, x2, Z)
    begin
        case State is
            when a0=>if Z='1' then NextState <= a1;
                    else NextState <= a0;
                    end if;
            when a1=>if x1='1' then NextState <= a2;
                    elsif x2='1' then NextState <= a4;
                    else NextState <= a3;
                    end if;
            when a2=>if x2='1' then NextState <= a4;
                    else NextState <= a3;
                    end if;
            when a3=>if x1='1' then NextState <= a5;
                    else NextState <= a0;
                    end if;
            when a4=>if x1='1' then NextState <= a5;
                    else NextState <= a0;
                    end if;
            when a5=>NextState <= a0;
            when others => NextState <= a0;
        end case;
    end process;
    y1 <= '1' when State=a1 else '0';
    y2 <= '1' when State=a2 else '0';
    y3 <= '1' when State=a3 else '0';
    y4 <= '1' when State=a4 else '0';
    y5 <= '1' when State=a5 else '0';
end;

```

Рисунок 4.2 – VHDL-модель, яка побудована по графу переходів МПА

На рис. 4.3 (а) наведені ТПВ зі стовбцем Sh та граф переходів з додатковими дугами Sh (рис. 4.3 б) при організації обходу вершин графу за природною нумерацією (перший варіант: a0–a1–a2–a3–a4–a5–a0), а на рис. 4.4 – другий варіант (a0–a1–a2–a4–a3–a5–a0) з урахуванням складності функцій збудження та оцінок тестопридатності.

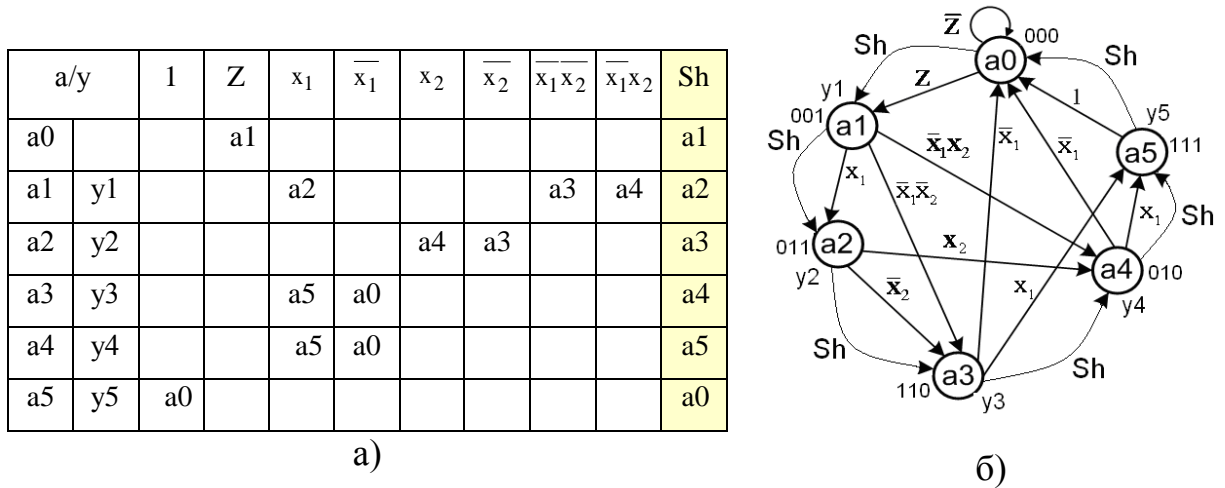


Рисунок 4.3 – Перший варіант ТПВ тестопридатного керуючого МПА (а) та його граф переходів (б)

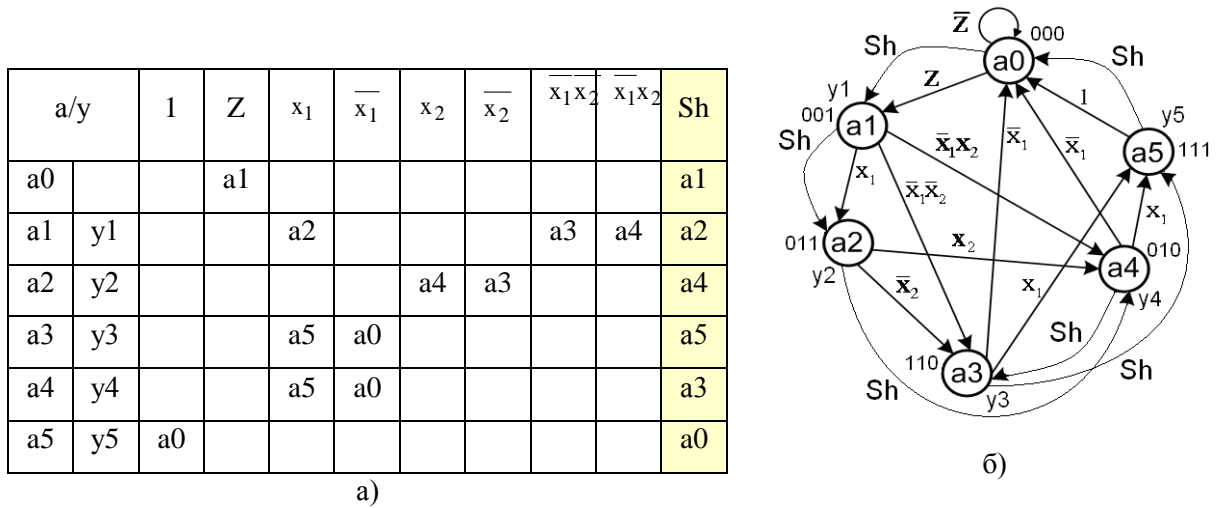


Рисунок 4.4 – Другий варіант ТПВ тестопридатного керуючого МПА (а) та його граф переходів (б)

Розглянемо VHDL-моделі тестопридатних МПА з різним способом розміщення дуг Sh (рис.4.3 та 4.4)) та виконаємо їх автоматизований синтез. Для скорочення місця на рис. 4.5 представлена архітектура VHDL-моделі

```

architecture Moore of FSM is
    type State_type is (a0, a1, a2, a3, a4, a5);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset)
    begin
        if Reset='1' then State <= a0;
        elsif Clk'event and Clk = '0' then State <=
NextState;
        end if;      end process;
        Sreg0_NextState: process (State, x1, x2, Z, Sh)
    begin
        case State is
            when a0=> if sh='1' then NextState <= a1;
                elsif Z='1' then NextState <= a1;
                else NextState <= a0;
                end if;
            when a1=> if sh='1' then NextState <= a2;
                elsif x1='1' then NextState <= a2;
                elsif x2='1' then NextState <= a4;
                else NextState <= a3;
                end if;
            when a2=> if sh='1' then NextState <= a3; -- (a4)
                elsif x2='1' then NextState <= a4;
                else NextState <= a3;
                end if;
            when a3=> if sh='1' then NextState <= a4; -- (a5)
                elsif x1='1' then NextState <= a5;
                else NextState <= a0;
                end if;
            when a4=> if sh='1' then NextState <=a5; -- (a3)
                elsif x1='1' then NextState <= a5;
                else NextState <= a0;
                end if;
            when a5=> if sh='1' then NextState <= a0;
                else NextState <= a0;
                end if;
            when others => NextState <= a0;
        end case;
    end process;
    y1 <= '1' when State=a1 else '0';
    y2 <= '1' when State=a2 else '0';
    y3 <= '1' when State=a3 else '0';
    y4 <= '1' when State=a4 else '0';
    y5 <= '1' when State=a5 else '0';
end;

```

Рисунок 4.5 – Архітектура VHDL-моделі тестопридатного МПА КА

тестопридатного МПА для першого варіанту розміщення дуг Sh (a0–a1–a2–a3–a4–a5–a0) та на ній же вказані зміни в послідовності станів Sh, що характерні для другого варіанту обходу графу (a0–a1–a2–a4–a3–a5–a0).

На рис. 4.6 наведена часова діаграма (Waveform) моделювання ДЕ по обходу станів КА за першим варіантом розміщення додаткових дуг Sh.

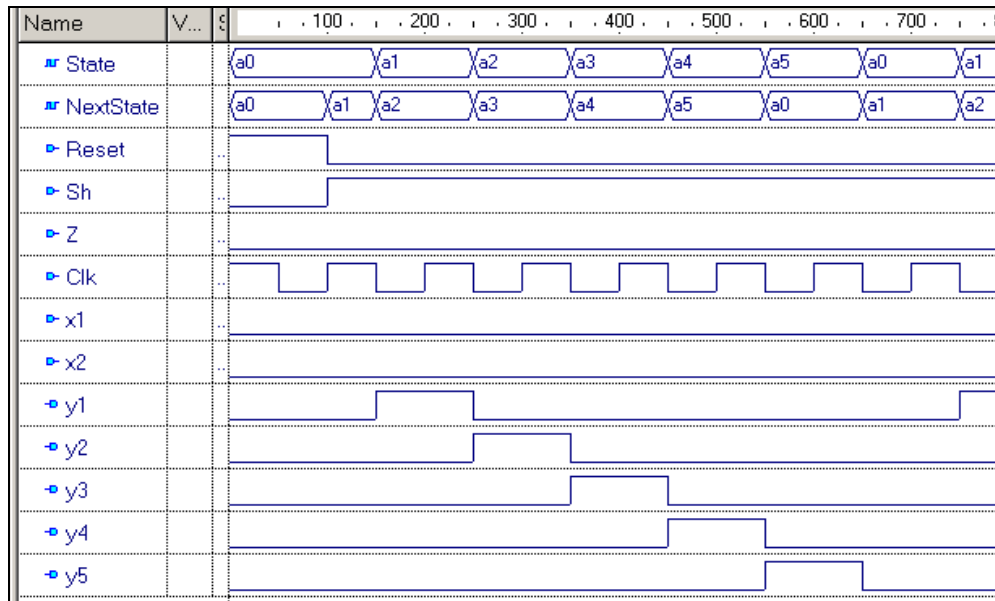


Рисунок 4.6 – Часова діаграма моделювання ДЕ по обходу станів КА

На рис. 4.7 наведений протокол автоматизованого синтезу інструментальними засобами САПР XILINX ISE зазначених VHDL-моделей. Відмітимо, що кодування станів для всіх варіантів є незмінним.

Analyzing FSM <FSM_0> for best encoding.

Optimizing FSM <State/FSM> on signal <State[1:3]> with gray encoding.

State | Encoding: a0|000 a1|001 a2|011 a3|110 a4|010 a5|111.

Рисунок 4.7 – Протокол автоматизованого синтезу

Повний протокол автоматизованого синтезу наведений у додатку Ж.

В аналітичному вигляді функція збудження триггеру першого розряду при першому варіанті розміщення дуг Sh буде:

$$D_1 = (Q_1 \overline{Q_3} \vee \overline{Q_1} Q_2 \overline{Q_3}) x_1 \overline{Sh} \vee \overline{Q_1} Q_2 \overline{Q_3} x_2 \overline{Sh} \vee (\overline{Q_1} Q_2 \overline{Q_3} \vee \overline{Q_1} Q_2 Q_3) Sh \vee \overline{Q_2} Q_3 x_1 x_2 \overline{Sh},$$

а для другого варіанту :

$$D_1 = (Q_1 \overline{Q_3} \vee \overline{Q_1} Q_2 \overline{Q_3}) x_1 \overline{Sh} \vee \overline{Q_1} Q_2 \overline{Q_3} x_2 \overline{Sh} \vee (Q_1 \overline{Q_3} \vee \overline{Q_1} Q_2 \overline{Q_3}) Sh \vee \overline{Q_2} Q_3 x_1 x_2 \overline{Sh},$$

тобто зменшення буде на 1 вхід вентиля та 1 інвертор, або на 4 % при оцінці по Квайну.

Загальні апаратурні витрати по Квайну при синтезі для схеми без Sh - 133, при першому варіанті схеми з Sh за умови кодування станів КА, яке формує САПР, - 187, а при другому варіанті – 184, що показує збільшення апаратурних витрат відповідно на 40% та 38%. Проведений аналіз показує, що при обранні варіанту розміщення дуг Sh слід враховувати не тільки складність функції збудження, а й кодування станів автомату.

4.2 Програмний модуль побудови тестопридатних HDL-моделей

В рамках реалізації розроблених моделей та процедур створено програмний модуль з візуальним інтерфейсом, який дозволяє будувати state diagram і отримувати простий та легкотестований HDL-код керуючого автомата на основі state diagram.

Однією із задач у циклі проектування цифрових пристроїв є побудова HDL-моделей, їх моделювання інструментальними засобами САПР. Виходячи з цього, створений хмарний сервіс, який забезпечить полегшення процесу роботи з VHDL-моделюванням, так як не дивлячись на активне використання програмованих інтегрованих схем, у мережі Інтернет (включаючи іноземні ресурси) до цього часу немає сервісів, які є розширеними версіями програмного продукту State Editor у складі Active-HDL. Зручність подібних сервісів як раз у тому, що користувачеві не потрібно встановлювати спеціальне програмне забезпечення для досягнення

виконання невеликих проектів. Специфіка популярних Web-мов (HTML, CSS і JavaScript) з використанням нових технологій дозволяє якісно запрограмувати багато можливостей, які властиві інтерфейсу Active-HDL.

Візуальний інтерфейс дозволяє будувати state diagram і отримувати готовий HDL-код, який описує поведінку наданого графа. Зручність подібних сервісів полягає у тому, що користувачеві не потрібно встановлювати спеціальне програмне забезпечення для досягнення виконання вказаних задач.

Переваги розробленого програмного продукту наступні:

- відсутність інсталяції;
- потреба лише в Internet-браузері, який підтримує javascript;
- можливість автоматичної побудови легкотестованого VHDL-коду;
- швидкість отримання коду;
- програмний продукт може бути реалізований у вигляді хмарного сервісу.

Сервісні функції системи:

- наявність графічного редактора для побудови графової моделі (технологія canvas);
- обробка вхідних даних у JS-контролері з результируючою генерацією VHDL-коду (з використанням мови Javascript);
- збереження HDL-коду у форматі *.vhd;
- можливість збереження моделі у форматі Json.

При побудові програмного продукту були використані наступні інструментальні засоби:

- сучасний браузер (Chrome);
- мова програмування Javascript;
- редактор коду Sublime 2;
- бібліотека jQuery;
- функції Canvas;

– формат вихідних даних – мова опису апаратури VHDL (код HDL-моделі у форматі *.vhd).

Особливості програмної реалізації та функціонування програмного продукту:

- кожна вершина та перехід представлені окремим об'єктом;
- у контролері проходить парсинг вхідних даних та їх структуризація;
- модель змінює свій стан залежно від даних та посилає їх на окремий блок інтерфейсу;
- готовий VHDL-код доступний для збереження завдяки окремій js-функції.

На рис. 4.8 показана загальна інформаційна структура розробленого програмного продукту. При цьому використовуються бібліотеки та шаблони мови програмування JavaScript.

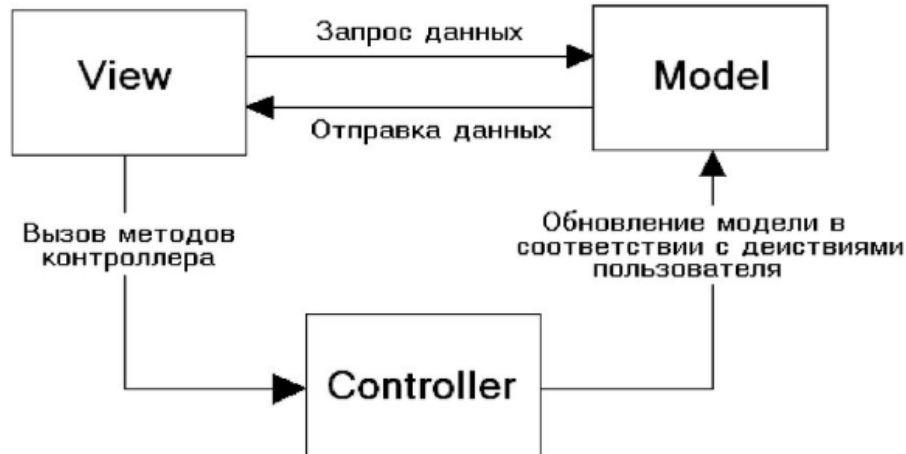


Рисунок 4.8 – Інформаційна структура програмного продукту.

У загальному вигляді логіка реалізації програмного забезпечення ділиться на два основні етапи:

- графічна побудова моделі;
- отримання готового VHDL-коду.

У свою чергу на другому етапі ми можемо отримати стандартний код для побудованої схеми або його легкотестований еквівалент. Логіка отримання легкотестованого коду реалізована за допомогою додавання додаткового символу Sh в таблицю переходів.

Графічна побудова моделі виконана за допомогою технології canvas (технології для створення 2d зображень за допомогою інструментів javascript) та спеціальним функціям та модулям (рис. 4.9).

```

/*function changing_func(ev) {
    var current_value = $(ev).val();
    if(current_value == 1){
        $(ev).parent().siblings('.elements-
destination').find('.x-coord').val("0");
    }
    else if(current_value == 0){
        $(ev).parent().siblings('.elements-
destination').find('.x-coord').val("1");
    }
}*/
var libs = '';
function libs_choose() {
    if(libs.indexOf($('#libs').val()+1) == 0) {
        libs += '\n      '+$('#libs').val();
        $('#libraries').val(libs);
    }
}
jQuery(document).ready(function( $ ){
    $('#password_accept').click(function() {
        if($('#password').val() == "diplom") {
            $('#security').remove();
        }
    })
    $('#automate_size').click(function() {
        custom_value();
    })
    $('#security').hide();
    var x = 0;
    var offset;
    function showCoords(event) {

```

Рисунок 4.9 – Функції графічного вводу Javascript

Більш детальний розгляд логіки програми виглядає наступним чином: інформація, яка була введена за допомогою графічного редактору відправляється на модуль парсингу. Там вхідні дані структуруються та оброблюються відповідно до стандартів та синтаксису VHDL. Кожен процес

та цикл обробляються окремо, за допомогою спеціальних функцій та умов.

Для налаштування візуального інтерфейсу та побудові state diagram керуючого автомату потрібно задати назву проекту (Entity) архітектури, встановити значення для скидання у початковий стан (reset), тип синхронізації тощо. На рис. 4.10 (а) представлений візуальний інтерфейс налаштування параметрів VHDL-моделі. Програмою передбачена можливість вводу як символічних імен станів автомату (найменувань вершин графу переходів) так і кодувань станів автомату у двійковому алфавіті (рис. 4.10 б).

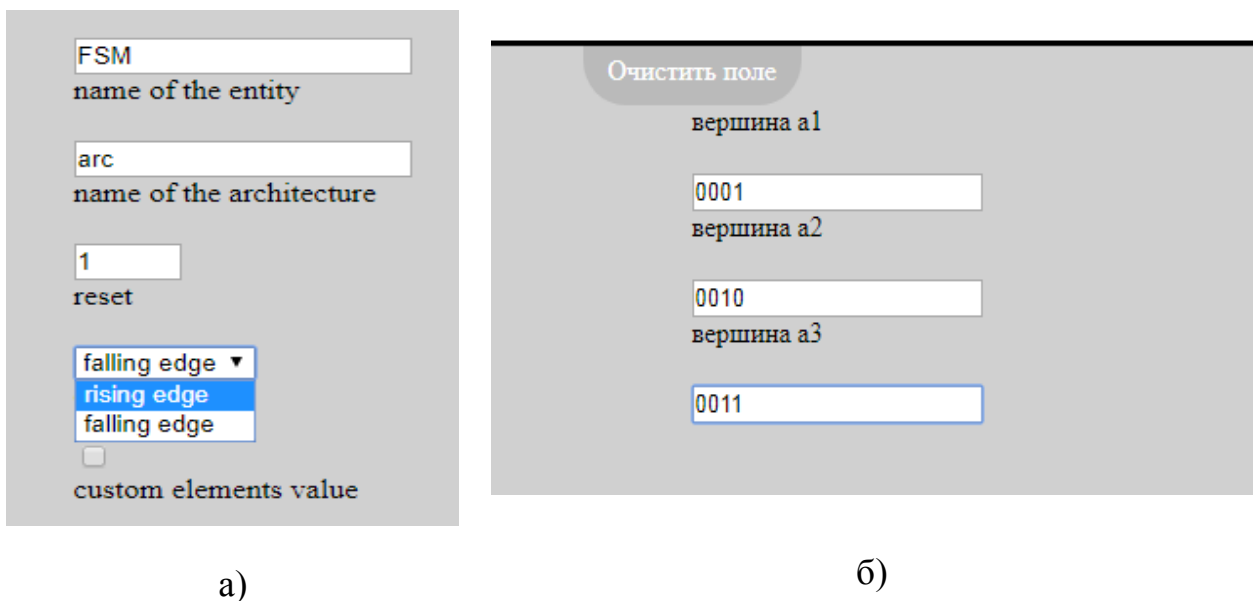


Рисунок 4.10 – Налаштування параметрів побудови VHDL-моделі

Введення вхідних даних та побудова схеми здійснюються за допомогою графічного інтерфейсу. У процесі побудови моделі використовуються вершини та дуги, що з'єднують ці вершини. Кожній вершині задається своє унікальне ім'я (зазвичай a_1, a_2, \dots, a_n , залежно від кількості вершин). Кожна дуга містить в собі інформацію про початкову та кінцеву вершину, умови переходу та вихідні сигнали. На рис. 4.11 наведений приклад вводу графу переходів КА системи управління контрольним пунктом у газорегуляторного пункту. Коли всі необхідні дані внесені, надається можливість сгенерувати VHDL-код для цього графу переходів КА (рис. 4.12).

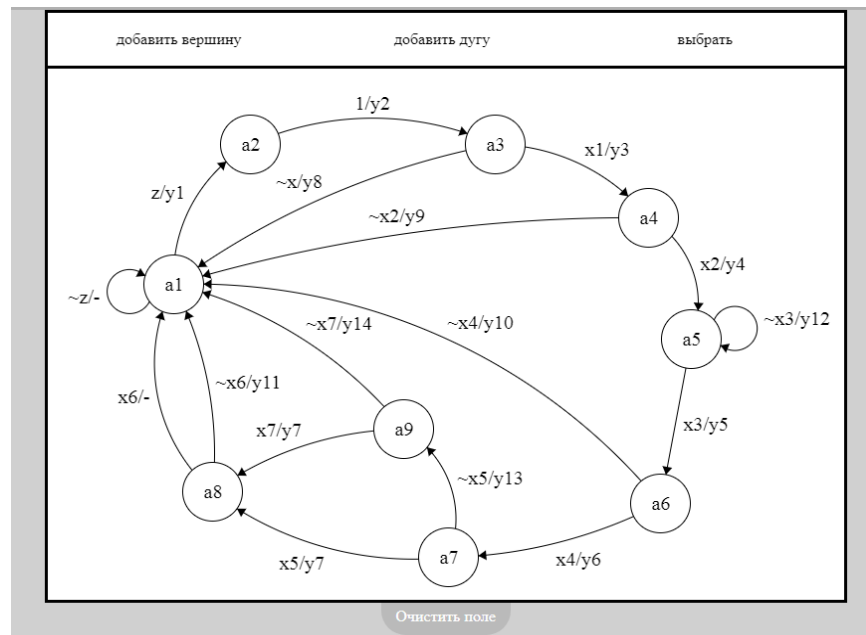


Рисунок 4.11 – Графічний інтерфейс при побудові графої моделі КА КП ГРП

```

architecture arc of FSM is
type statetype is (a1, a2, a3, a4, a5, a6, a7, a8, a9);
signal state, nextstate: statetype;
begin
  State_NextState: process (X, State)
  begin
    case state is
      when a1 =>
        if (X(0)='J') then nextstate <= a1;
        else nextstate <= a2; y1 <= 1;
        end if;
      when a2 => nextstate <= a3; y2 <= 1;
      when a3 =>
        if (X(1)='J') then nextstate <= a1; y8 <= 1;
        else nextstate <= a4; y3 <= 1;
        end if;
      when a4 =>
        if (X(2)='J') then nextstate <= a1; y9 <= 1;
        else nextstate <= a5; y4 <= 1;
        end if;
      when a5 =>
        if (X(3)='J') then nextstate <= a5; y12 <= 1;
        else nextstate <= a6; y5 <= 1;
        end if;
      when a6 =>
        if (X(4)='J') then nextstate <= a1; y10 <= 1;
        else nextstate <= a7; y6 <= 1;
        end if;
      when a7 =>
        if (X(5)='1') then nextstate <= a8; y7 <= 1;
        else nextstate <= a9; y <= 0;
    end case;
  end process;
end;

```

Сгенерировать код Сгенерировать SH код Сохранить в .vhd

Рисунок 4.12 – VHDL-модель КА в системі управління КП ГРП

Приклад моделювання вхідної послідовності при проведенні ДЕ по обходу послідовності станів (a1 – a2 – a3 – a4 – a5 – a6 – a1) для згенерованої VHDL-моделі наведено на рис. 4.13.

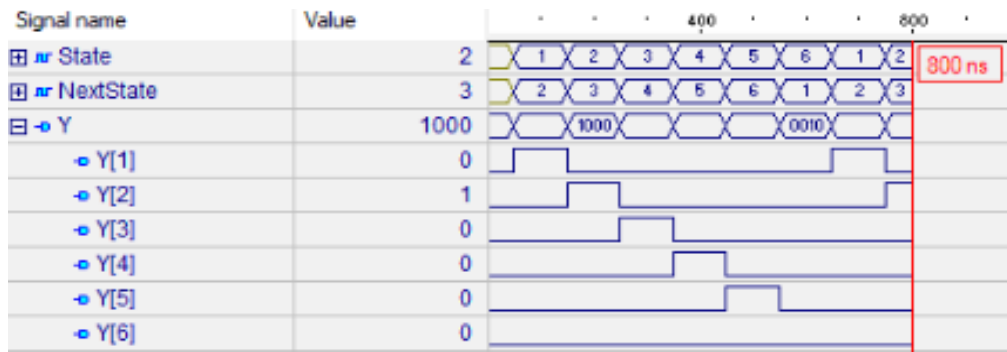


Рисунок 4.13 – Результати моделювання VHDL-моделі КА КП ГРП

4.3 Тестопридатне проектування системи управління КП ГРП

Одним із способів розширення вхідного алфавіту для вирішення завдання підвищення тестопридатності є введення додаткового стовпця в ТПВ, що відповідає додатковій дузі в графі переходів і додатковому рядку в HDL-кодi. Розширення ТПВ автомата шляхом додавання стовпця Sh (Shift) підвищує керованість станів автомата, робить його легкотестованим і забезпечує установку автомата в будь-який заданий стан за $(n-1)$ тактів.

В результаті отримана VHDL-модель (рис. 4.14, 4.15) легкотестованої системи автоматичного регулювання КП ГРП, що дозволяє проводити процес тестування в польових умовах на видаленій території без присутності кваліфікованого персоналу і персонального комп'ютера. Уся контрольна інформація виводиться на цифрове індикаційне табло. Побічним результатом роботи цього програмного модуля є отримання у довідковому режимі ТПВ КА системи управління КП ГРП (рис. 4.16).

Розглянемо варіант обходу послідовності станів $a_1-a_2-a_3-a_4-a_5-a_6-a_1$ за наявності сигналу Sh (рис. 4.17).

Відмітимо, що при розміщенні додаткових дуг Sh використовувався евристичний метод. На рис. 4.18 показані результати моделювання розробленої VHDL-моделі тестопридатного КА Мілі для КП ГРП по обходу послідовності станів $a_1-a_2-a_3-a_4-a_5-a_6-a_1$ за наявності сигналу Sh.

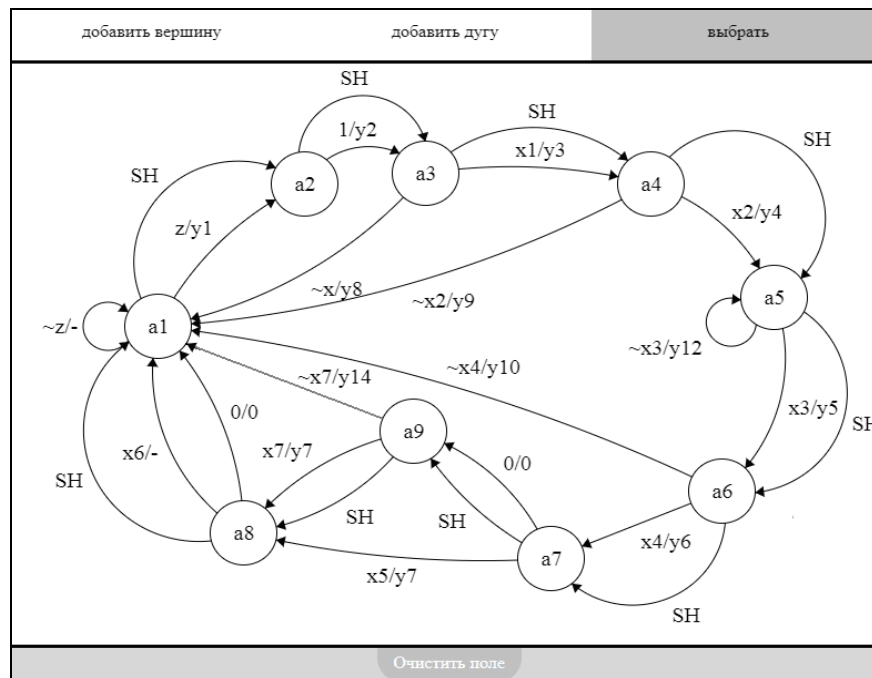


Рисунок 4.14 – Графичний інтерфейс побудови легкотестованої моделі КА

```

architecture arc of FSM is
type statetype is (a1, a2, a3, a4, a5, a6, a7, a8, a9);
signal state, nextstate: statetype;
begin
  State_NextState: process (X, State)
  begin
    case state is
      when a1 =>
        if ($h='1') then nextstate <= a2;
        elseif (X(0)='0') then nextstate <= a1;
        else nextstate <= a2; y1 <= 1;
        end if;
      when a2 =>
        if ($h='1') then nextstate <= a3;
        else nextstate <= a3; y2 <= 1;
        end if;
      when a3 =>
        if ($h='1') then nextstate <= a4;
        elseif (X(1)='0') then nextstate <= a1; y8 <= 1;
        else nextstate <= a4; y3 <= 1;
        end if;
      when a4 =>
        if ($h='1') then nextstate <= a5;
        elseif (X(2)='0') then nextstate <= a1; y9 <= 1;
        else nextstate <= a5; y4 <= 1;
        end if;
      when a5 =>
        if ($h='1') then nextstate <= a6;
        elseif (X(3)='0') then nextstate <= a5; y12 <= 1;
        else nextstate <= a6;
        end if;
    end case;
  end process;
end;

```

Сгенерировать код Сгенерировать SH код

Рисунок 4.15 – VHDL-модель легкотестованого КА для КП ГРП

a/x	\bar{z}	z	1	x1	$\bar{x}1$	x2	$\bar{x}2$	x3	$\bar{x}3$	x4	$\bar{x}4$	x5	$\bar{x}5$	x6	$\bar{x}6$	x7	$\bar{x}7$	Sh
a1	a1/ -	a2/ y1																a2/ -
a2			a3/ y2															a3/ -
a3				a4/ y3	a1/ y8													a4/ -
a4						a5/ y4	a1/ y9											a5/ -
a5								a6/ y5	a5/ y12									a6/ -
a6										a7/ y6	a1/ y10							a7/ -
a7												a8/y 7	a9/ y13					a9/ -
a8														a1/ -	a1/ y11			a1/ -
a9																a8/ y7	a1/ y14	a8/ -

Рисунок 4.16 – ТПВ з додатковим стовпцем КА КП ГРП

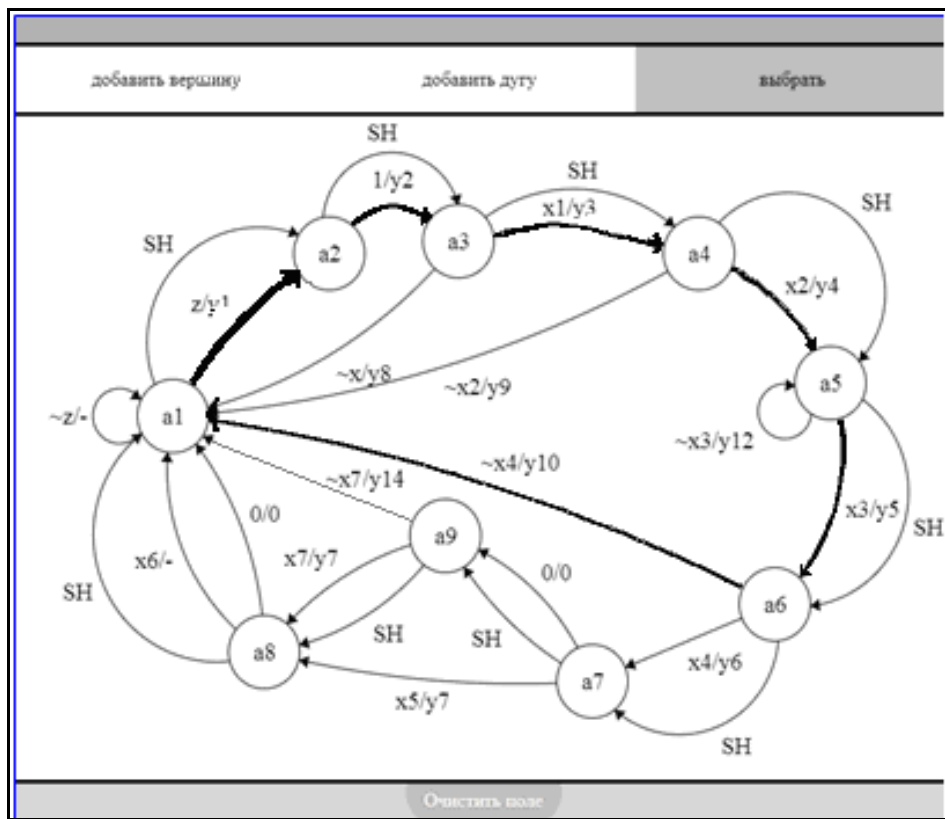


Рисунок 4.17 – VHDL-модель КА у системі управління КП ГРП

Signal name	Value				200		400		600
nr State	6	U	1	2	3	4	5	6	
nr NextState	1	U	2	3	4	5	6	1	
Y	00	00	00	00	00	00	00	00	
nr Sh	1								

Рисунок 4.18 – Результат моделювання VHDL-моделі КА для КП ГРП

Відсутність вихідних сигналів в режимі організації обходу графа переходів автомата пов'язана з тим, що вихідні сигнали y_i автомата Мілі пов'язані з дугами графа переходів, а введення додаткових дуг Sh не передбачає наявності якого-небудь вихідного сигналу (зважаючи на скорочення апаратних витрат).

В таблиці 4.1 наведені апаратні витрати на легкотестований КА в системі управління КП ГРП по результатам автоматизованого синтезу.

Таблиця 4.1 – Апаратні витрати на легкотестований КА

№	Обраний пристрій 3s500efg320-5	Апаратні витрати	Апаратні витрати с Sh
1	Кількість модулів	10 out of 4656	12
2	Кількість тригерів	9 out of 4656	12
3	Кількість 4-х входових LUTs	18 out of 9312	22
4	Number of IOs	24	25
5	Number of bonded IOBs	24 out of 232	25

Експерименти по схемній реалізації тестопридатних КА засобами САПР XILINX ISE на ПЛИС Spartan 3E показали, що додаткові апаратні витрати для синтезу тестопридатного керуючого автомата Мілі для КП ГРП в середньому не перевищують 25% від загальних апаратних витрат на основний керуючий автомат. Це підтверджує ефективність запропонованих підходів автоматизованого проектування легкотестованих керуючих автоматів.

Аналіз ефективності запропонованих методів забезпеченості тестопридатності КА здійснювався шляхом аналізу апаратних витрат для результатів синтезу за критерієм Квайна для різних типів КА та різних способів забезпечення додаткових переходів в графах КА, що показано на рис. 4.19. Порівняльні дані взяті з таблиць 3.1, 4.1 та схемної реалізації МПА у додатку Ж.

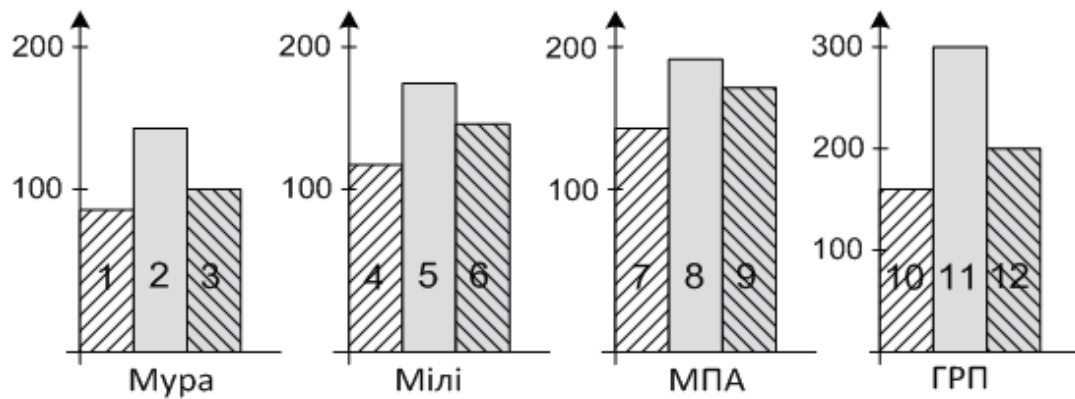


Рисунок 4.19 – Порівняльний аналіз апаратурних витрат

На рис.4.19 використані наступні позначення: 1 – апаратурні витрати (АВ) на простий КА Мура; 2 – АВ на тестопридатний КА Мура з зсувним регістром; 3 – АВ на легкотестований КА Мура з додатковим стовбцем Sh в ТПВ; 4 – АВ на простий КА Мілі; 5 – АВ на тестопридатний КА Мілі з зсувним регістром; 6 – АВ на легкотестований КА Мілі з додатковим стовбцем Sh в ТПВ; 7 – АВ на простий МПА Мура; 8 – апаратурні витрати на тестопридатний МПА Мура з природною послідовністю Sh; 9 – АВ на тестопридатний МПА Мура з оптимальною послідовністю Sh; 10 – АВ на простий КА в контурі регулювання ГРП; 11 – АВ на КА з пристроєм діагностування в контурі регулювання ГРП; 12 – АВ на легкотестований КА Мілі з додатковим стовбцем Sh в ТПВ в контурі регулювання ГРП.

Наведені результати показали достовірність та ефективність запропонованих методів внесення надлишковості в схемну реалізацію КА з метою забезпечення легкотестованості КА в системах логічного управління.

4.4 Висновки по розділу 4

1. Запропонований підхід до проектування тестопридатних (легкотестованих) КА в системах логічного управління. Тестопридатність досягається за рахунок введення апаратурної надлишковості на етапі

автоматизованого проектування КА. Легкотестованість досягається за рахунок забезпечення можливості установки КА в будь-який зазначений стан за $(n-1)$ тактів, де n – число станів КА. Діагностичні експерименти (тестування) КА виконуються шляхом реалізації різних способів обходу вершин і дуг графа переходів.

2. За результатами автоматизованого синтезу керуючого автомату з використанням САПР XILINX ISE показано, що вибір оптимального, з точки зору ваги функцій переходів, гамільтонового циклу у діагностичному експерименті обходу станів у графі КА дає мінімальні додаткові апаратурні витрати при розміщенні дуг Sh .

3. Розроблений програмний модуль з візуальним інтерфейсом для вводу графу переходів керуючого автомату та отримання його легкотестованої VHDL-моделі. Програмний модуль розроблений з використанням мови програмування JavaScript і може використовуватися як у локальному варіанті, так у вигляді хмарного сервісу.

4. Автоматизований синтез легкотестованої VHDL-моделі керуючого автомату в системі управління КП ГРП з використанням додаткових дуг Sh підтвердив, що цей підхід, по-перше, дає не більше 25% додаткових апаратурних витрат, а, по-друге, дає менші апаратурні витрати у порівнянні з іншими підходами підвищення тестопридатності (вбудований пристрій діагностування та додатковий регістр зсуву).

ВИСНОВОК

Проведені науково-технологічні дослідження в рамках дисертаційної роботи успішно вирішують актуальну науково-практичну задачу розробки моделей і методів автоматизованого проектування та діагностування тестопридатних цифрових систем логічного управління на основі кінцевих автоматів.

Автором одержано такі наукові та практичні результати:

1. Нова модель розширення вхідного алфавіту кінцевого автомата за рахунок введення додаткового стовпця в таблицю переходів-виходів автомата та додаткового входу в схемну реалізацію керуючого автомата, що дозволило за рахунок додаткових переходів встановлювати керуючий автомат у довільний стан за $(n - 1)$ тактів, де n – кількість станів керуючого автомата. Це суттєво скоротило довжину, час та точність методів проведення контролю і діагностики апаратної частини автоматних систем логічного керування.

2. Удосконалені методи проведення діагностичного експерименту з використанням синхронізуючих послідовностей, що дозволило повертати автомат в початковий стан навіть для автоматів з несправностями функцій переходів. Це дало можливість проводити неруйнівні діагностичні експерименти шляхом побудови гамільтонового циклу для невиключного класу керуючих автоматів, у яких функції виходів для різних станів можуть не відрізнятися.

3. Розвинені методи побудови апаратних пристроїв діагностування, що реалізують стратегію обходу всіх станів або дуг графа переходів керуючого автомата шляхом імітації алгоритму роботи операційного автомата в системі логічного управління. Це дозволило проводити процедури діагностування в рамках здійснення робіт з технічного обслуговування автоматних пристроїв керування, не порушуючи основний режим роботи критичної системи логічного управління на тривалий час.

4. Удосконалені методи розрахунку тестопридатності цифрових автоматів шляхом розрахунку досяжності вершин (станів) у графах переходів керуючих автоматів, які на відміну від існуючих методів, враховують

складність функцій переходів та виходів. Працездатність запропонованого методу підтверджено перевіркою результатів за допомогою математичного методу шляхів та гілок. Це дозволило мінімізувати додаткові апаратні витрати при побудові легкотестованих автоматів за рахунок введення додаткового входу сканування станів, що підтверджено оцінками по Квайну синтезованих моделей автоматів.

5. Новий підхід до автоматизованого проектування тестопридатних кінцевих автоматів за рахунок побудови моделей з використанням мов опису апаратури (HDL-моделей), у яких додаткові переходи реалізуються шляхом додавання умовних операторів до опису функцій переходів і автоматного шаблону мовою опису апаратури VHDL. Розроблені VHDL-описи моделюються та синтезуються інструментальними засобами систем автоматизованого проектування цифрових пристроїв XILINX ISE на технологічній платформі ПЛІС Spartan 3E.

6. Програмний модуль з візуальним інтерфейсом для введення графа переходів керуючого автомата та автоматизованої побудови тестопридатних HDL-моделей керуючих автоматів у формі автоматного шаблону у синтезованій підмножині мови опису апаратури VHDL. Програмний модуль розроблений з використанням мови програмування JavaScript і може використовуватися як у локальному варіанті, так і у вигляді хмарного сервісу. Автоматизований синтез легкотестованої VHDL-моделі КА в системі управління КП ГПП з використанням додаткових дуг Sh підтвердив, що цей підхід по-перше, дає не більше 25% додаткових апаратних витрат, а, по-друге, дає менші апаратні витрати у порівнянні з іншими підходами підвищення тестопридатності (вбудований пристрій діагностування та додатковий регістр зсуву).

Ринкова привабливість дослідження. Впровадження моделей та методів тестопридатного проектування критичних систем логічного управління на основі керуючих автоматів приведе до зменшення часу на проектування і технічне обслуговування автоматних систем логічного управління та кількості обслуговуючого персоналу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аврутов В.В. Надежность и диагностика приборов и систем / В.В. Аврутов, Н.И. Бурау. – К.: НТТУ «КПИ», 2014. – 156 с.
2. Шубин, Р.А. Надёжность технических систем и техногенный риск, учебное пособие / Р.А. Шубин. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2012. – 80 с.
3. IEC 61508 -1:2010 (EQV). Functional safety of electrical/electronic/programmable electronic safety-related systems. – Part 1: General requirements. – NSAI Standards, 2010. – 17 p.
4. ДСТУ 3433-95. Надійність техніки. Моделі відмов. Терміни та визначення. – Чинний від 1996-01-01. – К.: Держспоживстандарт України, 2001. – 45 с.
5. Малиновский М.Л. Управление объектами критического применения на основе ПЛИС: монография / М.Л. Малиновский. – Харьков : Факт, 2008. – 224 с.
6. Безопасность железнодорожной автоматики и телемеханики : Термины и определения / В. В. Сапожников, В. В. Сапожников, В. И. Талалаев, Д. В. Гавзов, Д. С. Марков // Автоматика, телемеханика и связь. – 1992. – № 4. – С. 30 - 32.
7. Отказобезопасные информационно-управляющие системы на программируемой логике / Бахмач Е. С., Герасименко А. Д., Головир В. А. и др.; под ред. Харченко В. С. – Х.: Нац. аэрокосм. ун-т им. Н. Е. Жуковского «ХАИ»; Науч.-произв. предприятие «Радий», 2008. – 380 с.
8. Avizienis A. Basic Concepts and Taxonomy of Dependable and Secure Computing / A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr // IEEE Trans. On Dependable and Secure Computing. – 2004. – Vol. 1. - № 1. – P. 11-13.
9. Методы построения безопасных микросистем железнодорожной автоматики / В. В. Сапожников, Вл. В. Сапожников, Х. А. Христов, Д. В. Гавзов ; под ред. В. В. Сапожникова. – М.: Транспорт.

1995. – 272 с.

10. Согомоян Е. С. Самопроверяемые устройства и отказоустойчивые системы / Е. С. Согомоян, Е. В. Слабаков. – М. : Радио и связь, 1989. – 207 с.

11. Сапожников В. В. Методы построения безопасных микроэлектронных систем железнодорожной автоматики / В. В. Сапожников, Вл. В. Сапожников, Х. А. Христов, Д. В. Гавзов ; под ред. Вл. В. Сапожникова. – М. : Транспорт, 1995. – 272 с.

12. Ubar R. Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source) / R. Ubar, J. Raik, H.-T. Vierhaus. – Information Science Reference. – Hershey ; New York : IGI Global, 2011. – 578 p.

13. Пархоменко П. П. Основы технической диагностики (оптимизация алгоритмов диагностирования, аппаратурные средства) / П. П. Пархоменко, Е. С. Согомоян. – М.: Энергоатомиздат, 1981. – 320 с.

14. Дрозд А. В. Рабочее диагностирование безопасных информационно-управляющих систем / А. В. Дрозд, В. С. Харченко, С. Г. Антощук, Ю. В. Дрозд, М. А. Дрозд, Ю. Ю. Сулима ; под ред. А. В. Дрозда, В. С. Харченко. – Харьков: Нац. аэрокосмич. ун-т им. Н. Е. Жуковского «ХАИ», 2012. – 614 с.

15. Ефанов Д. В. О свойствах кода с суммированием в схемах функционального контроля / Д. В. Ефанов, В. В. Сапожников, Вл. В. Сапожников // Автоматика и телемеханика. – 2010. – № 6. – С. 155–162.

16. Сапожников В. В. Организация функционального контроля комбинационных схем методом логического дополнения / В. В. Сапожников, Вл. В. Сапожников, А. В. Дмитриев, А. В. Морозов, М. Гессель // Электронное моделирование. – 2002. – Т. 24. – № 6. – С. 51–66.

17. Сідак В.С. Сучасні та інноваційні технології в безпеці газопостачання: монографія / В.С. Сідак, В.М. Супонєв, Ю.Ф. Броневський; за заг. ред. В.С. Сідака. – Харків: ХНУМГ ім. О.М. Бекетова, 2015. – 433 с.

18. Надежность и качество процессов регулирования современных

систем газоснабження: монографія / [В.С. Седак, В.Н. Супонев, Н.Д. Каслин и др.]. Под общ. ред. В.С. Сedaка, Харьк. нац. акад. город. хоз-ва – Х.: ХНАГХ, 2011. – 226 с.

19. Сідак В.С., Комплексні підходи до керування надійністю систем газопостачання: навчальний посібник / В.С. Сідак, О.С. Дудолад. – Харків: ХНАДУ, 2006.– 248 с.

20. Digital Transformation Initiative Oil and Gas Industry: White Paper of World Economic Forum [Електронний ресурс] / Reports weforum 2017. – Режим доступу: www / URL: <http://reports.weforum.org/digital-transformation/wp-content/blogs.dir/94/mp/files/pages/files/dti-oil-and-gas-industry-white-paper.pdf> – 05.12.2018 г.– Загол. з екрану.

21. Оценка и обеспечение качества программных средств космических систем / В. С. Харченко, В. В. Складар, Б. М. Конорев и др. ; под ред. Харченко В. С., Конорева Б. М.. – Х.: Нац. косм. агентство Украины; Гос. центр регулирования качества; Нац. аэрокосм. ун-т им. Н. Е. Жуковского «ХАИ», 2007. – 244 с.

22. Конорев Б.М. Инженерия критического программного обеспечения: учеб. пособие / Б.М. Конорев, И.Б. Туркин, С. В.Вергиенко, В.С. Харченко, Г.Н. Жолткевич, В.О. Мищенко, А.А.Гордеев— Х.: Нац. аэрокосм. ун-т им. Н. Е. Жуковского "ХАИ", 2015. – 68 с.

23. Ahmed S.U. How to design and test safety critical software systems / Syed Usman Ahmed, Muhammed Asim Azmi, Charu Badgujar// International Journal of Advances in Computer Science and Technology. – 2014.– Volume 3.– No.1 – P. 19-22.

24. Sommerville I. Software Engineering.- 10-th edition / Ian Sommerville. – Boston: Pearson Education Limited, 2016. – 816 p.

25. Майерс Г. Искусство тестирования программ. – 3-е изд.: пер. с англ. / Гленфорд Майерс, Том Баджетт, Кори Сандлер – М.: Издательство Вильямс, 2012 – 272 с.

26. Гилл А.. Введение в теорию конечных автоматов: перев. с англ. /

Артур Гилл. – М.: Издательство «Наука», 1966. – 272 с.

27. Миллер Р. Теория переключательных схем, т. II. Последовательностные схемы и машины: перев. с англ. / Реймонд Е. Миллер. – М.: Издательство «Наука», 1971. – 304 с.

28. Майоров С.А. Структура электронных вычислительных машин / С.А. Майоров, Г.И. Новиков. – Л.: Машиностроение, 1979. – 384 с.

29. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления / А.А. Шалыто. – СПб.: Наука, 1998. – 628 с.

30. Kulanov V. Fault-Tolerant SOPC-based Approaches with Multi-Version IP / V. Kulanov, V. Kharchenko, J. Prokhorova, S. Ostroumov // Радіоелектронні і комп'ютерні системи. – 2007. – №8 (27). – С. 71-77.

31. Малиновский М.Л. Математические модели безопасных ПЛИС-контроллеров с параллельной архитектурой / М. Л. Малиновский // Радиоэлектронные и компьютерные системы. – 2007. – №7. – С. 105-113.

32. Малиновский М. Л. Синтез безопасных автоматов с функциональной деградацией / М. Л. Малиновский // Управляющие системы и машины. – 2010. – № 1. – С. 84-91.

33. Barkalov A. Logic Synthesis for FSM-Based Control Units / Alexander Barkalov, Larysa Titarenko. – Berlin: Springer-Verlag Heidelberg, 2009. – 229 p.

34. Barcalov A. Logic Synthesis for Finite State Machines Based on Linear Chains of States / A. Barkalov, L. Titarenko, J. Bieganowski. – Berlin: Springer, 2017. – 225 p.

35. Мирошник М.А. Проектирование диагностической инфраструктуры вычислительных систем и устройств на ПЛИС: монография / М.А. Мирошник. – Х.: ХУПС, 2012. – 188 с.

36. Мирошник М.А. Метод проектирования строго безопасных автоматов локомотивной сигнализации / Л.В. Дербунович, Д.Г.Караман, М.Л. Малиновский, М.А. Мирошник, А.Н. Осипенко // Інформаційно-

керуючі системи на залізничному транспорті. – 2012. – №5. – С. 25-42.

37. Дербунович Л.В. Отказоустойчивые микроконтроллеры на основе сигнатурного мониторинга / Л.В. Дербунович, В.С. Суздаль, И.И. Тавровский, И.Н Темников // Информационно - управляющие системы на железнодорожном транспорте. – 2002. – №4, 5. – С. 71-73.

38. Siwar ben haj Hassin. Low Power Design Finite State Machine on Field Programmable Gate Arrays / Siwar ben haj Hassin, Mehdi Jemai, Bouraoui Oun // International Journal of Applied Engineering Research. – 2014. – V.9.– Num. 22.– P. 16341-16352.

39. Grzes T.N. Minimization of Power Consumption of Finite State Machines by Splitting Their Internal States / T.N. Grzes, V.V. Solov'ev // Journal of Computer and Systems Sciences International. – 2015. – Vol. 54 – No. 3. – P. 367–374.

40. Шалыто А.А. SWITCH-технология – автоматный подход к созданию программного обеспечения «реактивных» систем / А.А. Шалыто, Н.И. Туккель // Программирование. – 2001. – №5.– С.45-62.

41. Шалыто А.А. Логическое управление. Методы аппаратной и программной реализации. – СПб.: Наука, 2000. – 780 с.

42. Active-HDL Manual [Электронный ресурс] / Aldec Design Verification Company. – Режим доступа: [www / URL: https:// www.aldec.com /en /support/resources/documentation?type=2&products=&category=2](http://www.aldec.com/en/support/resources/documentation?type=2&products=&category=2) – 05.12.2018 г. – Загол. з екрану.

43. Зотов В.Ю. Практический курс сквозного проектирования цифровых устройств на основе ПЛИС фирмы Xilinx / В.Ю. Зотов // Современная электроника. – 2008.– № 3. – С. 64-71.

44. State Machine Diagram Tutorial [Электронный ресурс] / Visual Paradigm Online. – Режим доступа: [www / URL: https://online.visual-paradigm.com/tutorials/state-machine-diagram-tutorial/](http://www.visual-paradigm.com/tutorials/state-machine-diagram-tutorial/) – 10.11.2018 г.– Загл. с экрана.

45. EASE: State diagram editor [Электронный ресурс] / HDL Works. –

Режим доступа: [www / URL: https://www.hdlworks.com/products/ease/state_diagram.html](http://www.hdlworks.com/products/ease/state_diagram.html) – 10.11.2018 г.– Загл. с экрана.

46. Shamgunov N. State Machine Design Pattern / N. Shamgunov, G. Korneev, A. Shalyto // Shot communication papers conference proceedings of 4-th International Conference .NET Technologies in Central Europe 2006, May 29 – June 1 2006. – Net Technologies University of West Bohemia. – P. 51-57.

47. Шалыто А. А. Автоматное программирование / Н.И. Поликарпова, А.А. Шалыто. – 2-е изд.– Спб.: Питер, 2011. – 167 с.

48. Хаханов В.И. Проектирование и верификация цифровых систем на кристаллах. Verilog and System Verslog / В.И. Хаханов, И.В. Хаханова, Е.И. Литвинова, О.А. Гузь. – Харьков: ХНУРЭ, 2010. – 528 с.

49. Brown S. Fundamentals of Digital Logic with VHDL Design / S. Brown, Z. Vranesic. – 3rd. ed. – New York: McGraw-Hill Education., 2011. – 960 p.

50. Wisniewski R. Synthesis of Compositional Microprogram Control Units for Programmable Devices / Remigiusz Wisniewski. – Poland: University of Zielona Góra Press, 2009. – 153 p.

51. Pedroni V.A. Finite State Machines in Hardware: Theory and Design (with VHDL and SystemVerilog) / V.A. Pedroni. – Burlington: The MIT Press., 2013. – 352 p.

52. Grotker T. System Design with SystemC / T. Grotker, S. Liao, G. Martin, S. Swan. – Kluwer Academic Publishers, 2002.– 236 p.

53. IEEE Std. 1800–2005 IEEE Standard for SystemVerilog – Unified Hardware Design, Specification and Verification Language, 2005. – 648 p.

54. IEEE Std. 1076-2008 IEEE Standard VHDL Language Reference Manual, 2008. – 640 p.

55. IEEE Std. 1364-2005 IEEE Standard for Verilog hardware description language, 2006. – 560 p.

56. Haskell R. Digital Design Using Digilent FPGA Boards - VHDL / R. Haskell, D. Hanna. – 3rd. ed. – Rochester: LBE Books, 2009. – 395 p.

57. Zwolinski, M. Digital System Design with VHDL / M.Zwolinski – 2nd. ed. – New York: Pearson., 2004. – 384 p.
58. Taraate, V. PLD Based Design with VHDL: RTL Design, Synthesis and Implementation / V. Taraate. – Singapore: Springer, 2017. – 423 p.
59. Chu, P. P. FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version / P. P. Chu.– Cleveland: Wiley-Interscience, 2011. – 468 p.
60. Хаханов В.И Проектирование и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь.– Харьков: ХНУРЭ, 2009. – 484 с.
61. Young B. Digital Signal Integrity: Modeling and Simulation with Interconnects and Packages / B. Young. – Prentice Hall PTR, 2001. – 535 p.
62. Alur R. A theory of timed automata / R. Alur, D.L. Dill // Theoretical Computer Science. – 1994. – V.126. – N 2. – P. 183-235.
63. Грушвицкий Р.И. Проектирование систем на микросхемах программируемой логики / Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов.– СПб: БХВ-Петербург. – 2002. – 608 с.
64. Матюшин А.О. Программирование микроконтроллеров: Стратегия и тактика / А.О. Матюшин. – М.: ДМК Пресс, 2017. – 356 с.
65. Дианов В.Н. Диагностика и надежность автоматических систем: учебное пособие / В.Н. Дианов. – М.: Изд-во МГИУ, 2005. – 160 с.
66. Hennie E.G. Fault detection experiments for sequential circuits / E.G. Hennie // Proceeding of Fifth Symposium on Switching Circuit Theory and Logical Design – 1964. – P. 95 – 110.
67. Основы технической диагностики. Модели объектов, методы и алгоритмы диагноза / Под ред. П.П. Пархоменко.– М.: Энергия, 1976. – 460 с.
68. Тоценко В.Г. Алгоритмы технического диагностирования цифровых устройств / В.Г. Тоценко. – М.: Радио и связь, 1985. – 240 с.
69. Богомолов А.М. Контроль и преобразования дискретных автоматов / А.М. Богомолов, И.С. Грунский, Д.В. Сперанский. – К.: Наукова думка, 1975. – 175 с.

70. Abramovici M. Digital System Testing and Testable Design / M. Abramovici, M.A. Breuer and A.D. Friedman. – Comp. Sc. Press, 1998. – 652 p
71. Хаханов В.И. Система генерации тестов для проектирования цифровых автоматов в среде ACTIVE-HDL / В.И. Хаханов, Е.В. Ковалев, В.В. Ханько, Масуд М.Д. Мехеди // АСУ и приборы автоматики. – 2000. – Вып.111. – С. 15-22.
72. Шкиль А.С. Поиск ошибок проектирования в HDL-моделях цифровых автоматов / С. Альмадхоун, Е.Е. Сыревич, А.С. Шкиль // Вестник Херсонского НТУ. – 2013. – №2 (46). – С. 377-383.
73. Шкиль А.С. Автоматизация поиска ошибок проектирования в HDL-моделях конечных автоматов / А.С. Шкиль, А.С. Серокурова, Г.П. Фастовец // АСУ и приборы автоматики. – 2014. – Вып. 168. – С. 43-52.
74. Шкиль А.С. Диагностирование HDL-моделей микропрограммных автоматов / А.С. Шкиль, Э.Н. Кулак, А.С. Серокурова // АСУ и приборы автоматики. – 2015. – Вып. 172. – С. 22-31.
75. Шкиль А.С. Обнаружение ошибок проектирования в HDL-моделях конечных автоматов с использованием синхронизирующих последовательностей / А.С. Шкиль, М.А. Мирошник, Э.Н. Кулак А.С. Гребенюк, Д.Е. Кучеренко // Радиоэлектроника и информатика. – 2016. – № 3(74). – С. 39-46.
76. Куланов В. А. Инструментальное средство засева и моделирования константных неисправностей встроенных систем на ПЛИС/ В.А. Куланов, С.А. Куланов, А.С. Скрынник // Радіоелектронні і комп'ютерні системи. – 2010. – № 7. – С. 225-229.
77. Merayo M.G. Formal Testing from Timed Finite State Machines / M.G. Merayo, M. Nunez, I. Rodriguez // Computer Networks. – 2008. – Vol. 52–№2. – P. 432-460.
78. Zhigulin M., FSM-Based Test Derivation Strategies for Systems with Time-Outs / M. Zhigulin, N. Yevtushenko // Proc. of the 11th International Conference on Quality Software, QSIC2011. – IEEE, Madrid, 2011. – P. 141-149.

79. El-Fakih K. A practical approach for testing timed deterministic finite state machines with single clock / K. El-Fakih, N. Yevtushenko, A. Simao // *Science of Computer Programming*. – 2014. – Vol. 80. – P. 343–355.

80. Melnichenko I. An Extended Finite State Machine-Based Approach to Code Coverage-Directed Test Generation for Hardware Designs / I. Melnichenko, A. Kamkin, S. Smolov // *Proceedings of the Institute for System Programming*. – 2015. – № 27(3). – P. 161-182.

81. Backer J. A secure design-for-test infrastructure for lifetime security of SoCs / J. Backer; Sk Subidh Ali; K. Rosenfeld; D. Hély; O. Sinanoglu; R. Karri // *Proceedings IEEE International Symposium on Circuits and Systems (ISCAS'2015)* – 2015. – P. 37–40.

82. Zorian Y. Securing Test Infrastructure of System-on-Chips / G. Tshagharyan, G. Harutyunyan, S. Shoukourian, Y. Zorian // *Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2016)*. – 2016. – P. 29–32.

83. Zorian Y. Automated Flow for Test Pattern Creation for IPs in SoC / D. Sargsyan, G. Harutyunyan, S. Shoukourian, Y. Zorian // *Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2017)*. – 2017. – P. 21–24.

84. Dutt N. Toward Smart Embedded Systems: A Self-aware System-on-Chip (SoC) Perspective / Nikil Dutt, Axel Jantsch, Santanu Sarma // *ACM Transactions on Embedded Computing Systems*. – 2016 – Vol. 15. – No. 2, Article 22.–. P.2201-2227.

85. Беннеттс Р.Дж. Проектирование тестопригодных логических схем: пер. с англ. / Р. Дж. Беннеттс. – М.: Радио и связь, 1990. – 176 с.

86. Горяшко А.П. Проектирование легко тестируемых дискретных устройств: идеи, методы, реализация / А.П. Горяшко // *Автоматика и телемеханика*. – 1984. – № 7. – С. 5–35.

87. Stroud C.E. A designer's guide to built-in self-test / Charles E. Stroud. – Kluwer Academic Publishing, 2002 – 319 p.

88. Городецкий А. Введение в технологии JTAG и DFT. Тестирование в технологиях граничного сканирования и тестопригодное проектирование

/ Ами Городецкий. – Palmarium Academic Publishing, Germany, 2012 – 308 с.

89. Advanced HW Test Solutions [Электронный ресурс] / StarTest™. – Режим доступа: www / URL: <http://www.start-test.com> – 05.10.2018 г.– Загл. с экрана.

90. Шкиль А.С. Количественные оценки тестопригодности цифровых устройств на схмотехническом уровне / Г.Ф. Кривуля, А-М. Рами, А.С. Шкиль // Радиоэлектроника и информатика. – 1998. – №4. – С.75-79.

91. Соколов С.В. Оптимальні та адаптивні системи: навчальний посібник / С.В.Соколов. – Суми: Сум. держ. ун-т, 2018. – 220 с.

92. Хаханов В.И. Верификация цифровых устройств на основе использования анализа тестопригодности и ассерционных библиотек / В.И. Хаханов, М.А. Каминская, С.А. Зайченко // АСУ и приборы автоматики. – 2007. – Вып. 140. – С. 75-83.

93. Хаханов В.И. Модель облачного сервиса для поиска оптимального пути / С.В. Чумаченко, Е.И. Литвинова, В. И. Хаханов, А.Н. Зиарманд // Paradigmata poznání. Vědecko vydavatelské centrum «Sociosféra-CZ», s.r.o., Praha, Česká republika. – 2017. – Вып. 3. – С. 63-83.

94. Hahanov V. Cyber Physical Computing for IoT-driven Services / Vladimir Hahanov. – Springer International Publishing, 2018. – 279 p.

95. Asvini S. Design for Testability in Timely Testing of VLSI Circuits / S. Asvini, C. Nirmala // Int. Journal of Engineering Research and Applications. – 2015 – Vol. 5, – Issue 3, (Part -2). – P. 10-13.

96. Parzan F. Design for Testability of Sleep Convention Logic / Farhad A. Parsan, Scott C. Smith, Waleed K. Al-Assadi // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. – 2015 – Vol. 24. – P.743-753.

97. Gaur H. Testable Design of Reversible Circuits using Parity Preserving Gates / Hari Mohan Gaur, Ashutosh Kumar Singh, Umesh Ghakar // IEEE Design & Test. – 2018. – Vol. 35. – Issue 4. – P. 56-64.

98. Hosokawa T. Controller augmentation and test point insertion at RTL for concurrent operational unit testing / T. Hosokawa, S. Takeda, H. Yamazaki,

M. Yoshimura // Proc. of 2017 IEEE 23-rd International Symposium on On-Line Testing and Robust System Design (IOLTS'17), 3-5 July 2017. – P. 17-20.

99. Kuen-Jong L. On-Chip Self-Test Methodology With All Deterministic Compressed Test Patterns Recorded in Scan Chains / Lee Kuen-Jong, Chen Bo-Ren, Michael Andreas Kochte // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2018. – vol. 38. – № 2. – P. 309-321.

100. Kochte M. A.; Self-Test and Diagnosis for Self-Aware Systems / M.A. Kochte; H-J. Wunderlich // [IEEE Design & Test](#). – 2018. – V. 35. – [Iss. 5](#). –P.7-18.

101. Hu K. Testing of flow-based microfluidic biochips: Fault modeling test generation and experimental demonstration / [K Hu](#), [F Yu](#), [TY Ho](#), [K Chakrabarty](#) // IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. – Oct.2014.– vol. 33, – num. 10. – P. 1463-1475.

102. Supriya L. FPGA Implementation of BIST (Built in Self Test) Enabled UART for Real Time Interface Applications / L. Supriya, J. Lingaiah, G. Kalyan // International Journal of Science, Engineering and Technology Research (IJSETR). – 2015 – Volume 4. – Issue 7. – P. 2645-2647.

103. Ahlawat S. A High Performance Scan Flip-Flop Design for Serial and Mixed Mode Scan Test / S. Ahlawat, J. Tudu, A. Matrosova, V. Singh // IEEE Transactions on Device and Materials Reliability. – 2018. – 18 (№2). – P. 321-331.

104. Doulcier M. AES-Based BIST: Self-Test, Test Pattern Generation and Signature Analysis / M. Doulcier; M.-L. Flottes; B. Rouzeyre 4th IEEE International Symposium on Electronic Design, Test and Applications (delta 2008), 23-25 Jan. 2008. – IEEE, 2008. – P. 314-321.

105. Drozd A. Component-based safety-oriented on-line testing of digital systems / A. Drozd, V. Kharchenko, A. Siora, V. Sklyar // IEEE East-West Design & Test Symposium (EWDTS'2010), September 17-20, 2010. – Sankt-Petersburg, Russia, 2010. – P. 135-140.

106. . Drozd A.. The use of natural resources for increasing a checkability of the digital components in safety-critical systems / A. Drozd; V. Kharchenko; S. Antoshchuk; J. Drozd; M. Lobachev; J. Sulima // Proceedings of 10th IEEE

East-West Design & Test Symposium (EWDTS`2012), September 14-17, 2012. – Kharkov, Ukraine, 2012. – P. 283–288.

107. Drozd A., A method of the scan checking for on-line testing Of safe multithreaded bitwise pipeline system / A. Drozd, J. Drozd, V. Nikul, M. Al-Dhabi // *Радіоелектронні і комп'ютерні системи.*– 2016. – № 6 (80). – P.101-107.

108. Дрозд А. В. Методы рабочего диагностирования для цифровых компонентов систем критического применения / М.М. Аль-Даби, А.В. Дрозд, М.О. Дрозд, И.Н. Никоненко // *Холодильна техніка та технологія.*– 2017. – № 53(1). – P.85-88.

109. Drozd O. Hidden faults in FPGA-built digital components of safety-related systems / Oleksandr Drozd, Valeria Nikul, Viktor Antoniuk, Myroslav Drozd // *Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 20-24 Feb. 2018. – Slavske, Ukraine, 2018. – P. 805-809.*

110. Drozd, M. Safety-Related Instrumentation and Control Systems and a Problem of the Hidden Faults / M. Drozd, A. Drozd // *Digital Technologies: Proceedings of the 10th International Conference, 9-11 July 2014. – Zhilina, Slovak Republic. – P. 137-140.*

111. Баранов С.И. Синтез микропрограммных автоматов (граф схемы и автоматы).– 2-е изд., перераб. и доп./ С.И. Баранов.– Л.:Энергия, 1979. – 232 с.

112. Бережная М.А. Синхронизирующие последовательности в конечных детерминированных автоматах / М.А. Бережная (Мирошник) / *Вестник НТУ “ХПИ”.* – 2008. – № 57. – С. 7-15.

113. Макконнелл Дж. Анализ алгоритмов. Активный обучающий подход / Дж. Макконнелл – 3-е доп. издание. – Москва: Техносфера, 2009. – 416 с.

114. Харари Ф. Теорія графів / Ф. Харари. – М.: УРСС, 2003. – 296 с.

115. Пахомов Ю. В. Технологія ремонту газового обладнання і трубопровідних систем, монографія / І. І. Капцов, В. Г. Котух, Ю. В. Пахомов. – Харків : ХНУМГ ім. О. М. Бекетова, 2016. – 232 с.

116. Пахомов Ю. В. Проведення діагностичних експериментів у керуючих автоматах з використанням синхронізуючих послідовностей / О. С. Шкіль, М. А. Мірошник, Т. Г. Рожнова, Ю. В. Пахомов, Д. Г. Караман // Радіоелектроніка та інформатика: наук.-техн. журнал. – Харків: ХНУРЕ, 2018. – № 3 – С. 82-89.

117. Pakhomov Y. Design automation of easy-tested digital finite state machines / M. A. Miroshnik, Y. V. Pakhomov, A. S. Shkil, E. N. Kulak, D. Y. Kucherenko // Radio Electronics, Computer Science, Control, the scientific journal, Zaporizhzhia National Technical University, 2018. – № 2. – С. 117-124.

118. Пахомов Ю. В. Аналіз апаратурних витрат при тестопридатному проектуванні керуючих цифрових автоматів / М. А. Мірошник, Ю. В. Пахомов, О. С. Шкіль, Е. М. Кулак, Д. Ю. Кучеренко, І. В. Філіппенко // Вісник СХУ ім. В. Даля. – 2018. – № 6. – С. 101-109.

119. Pakhomov Y. V. Model of automated hardware diagnostics of remote energy systems management points / M. A. Miroshnyk, Y. V. Pakhomov // Світлотехніка та електроенергетика: міжнар. наук.-техн. журнал. – Харків: ХНУМГ ім. О. М. Бекетова, 2017. – № 3. – С. 3-9.

120. Пахомов Ю. В. Методы проектирования самопроверяемых цифровых автоматов / М. А. Миррошник, Э. Н. Кулак, Е. М. Алиева, Д. Г. Караман, Ю. В. Пахомов // Радиотехника: всеукр. межвед. науч.-техн. зб. – Харків: ХНУРЕ, 2016. – № 187. – С. 124-131.

121. Пахомов Ю. В. Методы синтеза легкотестируемых цифровых автоматов / М. А. Миррошник, Ю. В. Пахомов, А. С. Гребенюк И. В. Филиппенко // Інформаційно-керуючі системи на залізничному транспорті: наук.-техн. журнал. – Харків: УкрДУЗТ, 2016. – № 5. – С. 28-39.

122. Pakhomov Yuriy. Design of Logical Control Units Based on Finite State Machines' Patterns / Maryna Miroshnyk, Sergii Poroshyn, Alexander Shkil, Elvira Kulak, Inna Filippenko, Dariia Kucherenko, Yuriy Pakhomov, Salfetnikova Juliia, Maksym Goga // Proceedings of the 2018 IEEE East-West Design & Test Symposium, 14-17 Sept. 2018, – 6 p. [Електронний ресурс] / IEEE Xplore

Digital Library – Режим доступу: [www / URL: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?filter=issueId%20EQ%20%228524135%22&refinements=Author:Alexander%20Shkil&pageNumber=1&resultAction=REFINE](http://www.ieeeexplore.ieee.org/xpl/mostRecentIssue.jsp?filter=issueId%20EQ%20%228524135%22&refinements=Author:Alexander%20Shkil&pageNumber=1&resultAction=REFINE)– 10.11.18 г.

123. Pakhomov Y.V. Microware multiport multimeter sensor mutual reflection and its influence on signal and tract parameter measurement accuracy / M. A. Miroshnyk, Y. N. Salfetnikova, O.B. Zaichenko, N.Y. Zaichenko, Y.V. Pakhomov, A. N. Miroshnyk // XXVIII International Scientific Symposium "Metrology and Metrology Assurance". – 2018. – (September 10-14, Sozopol, Bulgaria) – P 60-64.

124. Пахомов Ю. В. Проектирование логических блоков управления с помощью шаблонов описания конечных автоматов / М. А. Мирошник, Ю. Н. Салфетникова, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: матеріали 31-ї міжнар. наук.- практ. конф. (Харків, УкрДУЗТ, 24–26 жовт. 2018р.). – № 4 – С. 21-22.

125. Pakhomov Yuri Design automation of testable finite state machines / Maryna Miroshnyk, Yuri Pakhomov, Edward German, Alexander Shkil, Elvira Kulak, Dariia Kucherenko // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2017), Kharkov National University of Radioelectronics, Novi Sad, Serbia, September 27– October 2 – 2017. – P 203-208.

126. Пахомов Ю. В. Исследование методов синтеза легкотестируемых цифровых устройств и систем / М. А. Мирошник, В. А. Крылова, Ю. В. Пахомов, А. Н. Мирошник // Проблеми інформатики та моделювання (ПІМ-2017): тези сімнадцятої міжнар. наук.-техніч. конф. (Харків, НТУ «ХПІ», 11–15 вер. 2017р.) – С. 58-59.

127. Пахомов Ю. В. Методы обнаружения ошибок проектирования в конечных автоматах с использованием синхронизирующих последовательностей / М. А. Мирошник, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: тези стендових доповідей та виступів учасників 30-ї міжнар. наук.- практ. конф. (Харків, УкрДУЗТ, 26–27 жовт. 2017р.). – № 4. – С. 15-16.

128. Пахомов Ю. В. Методи оцінки ефективності ремонтно-відновлювальних робіт для виробів газового обладнання та трубопровідних систем / М. А. Мірошник, В. Г. Котух, Ю. В. Пахомов. // Інформаційні технології та комп'ютерна інженерія: тези допов. п'ятої міжнар. наук.-практ. конф. (Івано-Франківськ, Вінниця, 27–29 трав. 2015р.). – С. 15-16.

129. Пахомов Ю. В. Структурный анализ измерительных систем датчиков для газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Інформаційні технології та комп'ютерна інженерія: тези допов. четвертої міжнар. наук.-практ. конф. (Вінниця, 28–30 трав. 2014р.). – С. 32-34.

130. Пахомов Ю. В. Системный подход к определению состава вспомогательного материала для изделий газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: тези стендових доповідей 27-ї міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 24–26 вер. 2014р.). – № 4. – С. 30-31.

131. Pakhomov Y. Information model of registration and analysis of technological factors arising during final processing of products of transport pipeline systems / V. Kotukh, N. Kaptsova, Y. Pakhomov, V. Kosenko // International Journal of Engineering and Technology (UAE, ISSN: 2227-524X). – 2018. – № 2.23. – С. 73-76.

132. Пахомов Ю. В. Методы автоматизации проектирования легкотестируемых компьютерных систем и устройств на основе цифровых автоматов / М. А. Мірошник, Л. А. Клименко, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. – 2018. – №4 (131). – С. 3-10.

133. Pakhomov Y. Methods for Designing Self-Checking Digital Machines / M. Miroschnyk, E. Kulak, E. Aliyeva, D. Karaman, Yu. Pakhomov // Telecommunications and Radio Engineering, USA. – 2017. – № 15 – С. 1367-1377.

134. Pakhomov Y. Konstruktion of Distributed Information Management Syste^ms for Accounting and Control of Energy Consumption at the Example of Gas / V. Kotukh, Y. Pakhomov // Telecommunications and Radio Engineering, USA. – 2016. – № 7. – С. 631-641.

135. Пахомов Ю. В. Разработка алгоритма прогнозирования функционально-технического состояния изделий газового оборудования и трубопроводных систем / Ю. В. Пахомов // Вестник БГТУ им. В. Г. Шухова: науч.- теор. журнал. – 2015. – № 3. – С. 90-97.

136. Пахомов Ю. В. Основные принципы создания единой распределенной системы автоматического контроля и учета энергоресурсов на примере газовой отрасли / В. Г. Котух, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. – Харків: УкрДУЗТ, 2015. – № 3. – С. 48-55.

137. Пахомов Ю. В. Построение распределенных информационно-управляющих систем учета и контроля энергоресурсов на примере газовой отрасли / В. Г. Котух, Ю. В. Пахомов // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харків: ХНУРЕ, 2015. – № 182.– С. 65-72.

138. Пахомов Ю. В. Исследование распределения тепловых потоков при импульсной лазерной сварке корпусов датчиков для газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Вестник БГТУ им. В. Г. Шухова. – 2014. – № 5. – С. 96-101.

139. Пахомов Ю. В. Технологическая концепция создания АСУ ТП для объектов энергоснабжения на базе цифровых распределенных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харків: ХНУРЕ, 2014. – № 179. – С. 131-137.

140. Пахомов Ю. В. Методы расчета упругих элементов мембранно-балочного типа в датчиках для газового оборудования и трубопроводных систем / Ю. В. Пахомов // Комунальне господарство міст: наук.- техн. зб. – Харків: ХНУМГ ім. О. М. Бекетова, 2014. – № 116. – С. 64-66.

ДОДАТОК А

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Список публікацій здобувача, в яких відображені основні наукові результати дисертації:

1. Пахомов Ю. В. Технологія ремонту газового обладнання і трубопроводних систем, монографія / І. І. Капцов, В. Г. Котух, Ю. В. Пахомов. – Харків: ХНУМГ ім. О. М. Бекетова, 2016. – 232 с.

2. Пахомов Ю. В. Проведення діагностичних експериментів у керуючих автоматах з використанням синхронізуючих послідовностей / [М. А. Мірошник, О. С. Шкіль, Ю. В. Пахомов та ін.] // Радіоелектроніка та інформатика: наук.-техн. журнал. – Харків: ХНУРЕ, 2018. – № 3 – С. 82–89 (Реферується або індексується міжнародними наукометричними базами Index Copernicus (<http://journals.indexcopernicus.com/-p24787015.3.html>), Google Scholar, CiteFactor, NBUV, SIS, OAJI.net, Cyberleninka, OECSP, Scholar Steer, TIU Hannover, I2OR).

3. Pakhomov Y. Design automation of easy-tested digital finite state machines / [M. A. Miroshnik, Y. V. Pakhomov, A. S. Shkil et al.] // Radio Electronics, Computer Science, Control, the scientific journal, Zaporizhzhia National Technical University. – 2018. – № 2. – P. 117–124 (Реферується або індексується міжнародними наукометричними базами Thomson Reuters Web of Science (WoS), CiteFactor, COPAC, CrossRef, eLibrary.ru / РИНЦ, GENERAL IMPACT FACTOR, Google Scholar, Impactfactor.pl, Index Copernicus, Scholar Steer, SIS, SSM).

4. Пахомов Ю. В. Аналіз апаратурних витрат при тестопридатному проектуванні керуючих цифрових автоматів / [М. А. Мірошник, Ю. В. Пахомов, О. С. Шкіль та ін.] // Вісник СХУ ім. В. Даля. – 2018. – № 6. – С. 101–109 (Реферується або індексується міжнародними наукометричними базами eLibrary.ru, NBUV, WorldWideScience.org, ScienceDirect, Google Scholar, WorldCat, BASE, DOAJ, URAN).

5. Pakhomov Y. V. Model of automated hardware diagnostics of remote energy systems management points / M. A. Miroshnyk, Y. V. Pakhomov // Світлотехніка та електроенергетика: міжнародний науково-технічний журнал. – Харків: ХНУМГ ім. О. М. Бекетова, 2017. – № 3. – Р. 3–9 (Реферується або індексується міжнародними наукометричними базами Index Copernicus, Google Scholar, WorldCat).

6. Пахомов Ю. В. Методы проектирования самопроверяемых цифровых автоматов / [М. А. Мирошник, Э. Н. Кулак, Ю. В. Пахомов и др.] // Радиотехника: всеукр. межвед. науч.-техн. зб. – Харків: ХНУРЕ, 2016. – № 187. – С. 124–131 (Реферується або індексується міжнародними наукометричними базами Google Scholar, НБУВ, Elibrary.ru).

7. Пахомов Ю. В. Методы синтеза легкотестируемых цифровых автоматов / А. С. Гребенюк, М. А. Мирошник, Ю. В. Пахомов, И. В. Филиппенко // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. – Харків: УкрДУЗТ, 2016. – № 5. – С. 28–39 (Реферується або індексується міжнародними базами Google Scholar, РІНЦ, Elibrary.ru).

Результати, які засвідчують апробацію матеріалів дисертації:

8. Pakhomov Y. Design of Logical Control Units Based on Finite State Machines' Patterns / [M. Miroshnyk, Y. Pakhomov, A. Shkil et al.] // IEEE East-West Design & Test Symposium (EWDTS'2018, Kazan, Russia, 14-17 Sept. 2018) – 6 p (Indexed by Scopus).

9. Pakhomov Y. V. Microware multiport multimeter sensor mutual reflection and its influence on signal and tract parameter measurement accuracy / [M. A. Miroshnyk, O. B. Zaichenko, Y. V. Pakhomov et al.] // 28-th International Scientific Symposium "Metrology and Metrology Assurance" (Sozopol, Bulgaria, 10–14 Sept. 2018). – Р. 60–64 (Реферується або індексується міжнародними наукометричними базами Elibrari.ru, Erih plus, Scince Index, DOAJ, РІНЦ).

10. Пахомов Ю. В. Проектирование логических блоков управления с помощью шаблонов описания конечных автоматов / М. А. Мирошник, Ю. Н. Салфетникова, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: 31-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 24–26 жовт. 2018р.). – № 4. – С. 21–22 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, Elibrary.ru).

11. Pakhomov Y. Design automation of testable finite state machines / [M. Miroshnyk, Y. Pakhomov, S. Shkil et al.] // IEEE East-West Design & Test Symposium (EWDTS'2017, Novi Sad, Serbia, Sept. 27–Oct. 2, 2017). – P 203-208 (Indexed by Scopus, IEEE Xplore).

12. Пахомов Ю. В. Исследование методов синтеза легкотестируемых цифровых устройств и систем / М. А. Мирошник, В. А. Крылова, Ю. В. Пахомов, А. Н. Мирошник // Проблеми інформатики та моделювання (ПІМ-2017): 17-та міжнар. наук.-техніч. конф. (Харків, НТУ «ХП», 11–15 вер. 2017р.) С. 58–59 (Реферується або індексується міжнародними наукометричними базами Index Copernicus, Google Scholar, OAJ, Cyberleninka, WorldCat, BASE, UIF, DOAJ, NBUV MIAR, DOI, CrossRef).

13. Пахомов Ю. В. Методы обнаружения ошибок проектирования в конечных автоматах с использованием синхронизирующих последовательностей / М. А. Мирошник, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: 30-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 26–27 жовт. 2017р.). – № 4. – С. 15–16 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, Elibrary.ru).

14. Пахомов Ю. В. Методи оцінки ефективності ремонтно-відновлювальних робіт для виробів газового обладнання та трубопровідних систем / М. А. Мірошник, В. Г. Котух, Ю. В. Пахомов. // Інформаційні технології та комп'ютерна інженерія: 5-та міжнар. наук.-практ. конф. (Івано-Франківськ, Вінниця, 27–29 трав. 2015р.). – С. 15–16 (Реферується або

індексується міжнародними наукометричними базами Google Scholar, РІНЦ, DOAJ).

15. Пахомов Ю. В. Структурный анализ измерительных систем датчиков для газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Інформаційні технології та комп'ютерна інженерія: 4-та міжнар. наук.-практ. конф. (Вінниця, 28–30 трав. 2014р.). – С. 32–34 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, DOAJ).

16. Пахомов Ю. В. Системный подход к определению состава вспомогательного материала для изделий газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: 27-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 24–26 вер. 2014р.). – № 4. – С. 30–31 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, Elibrary.ru).

Публікації, які додатково відображають наукові результати дисертації:

17. Pakhomov Y. Information model of registration and analysis of technological factors arising during final processing of products of transport pipeline systems / V. Kotukh, N. Kaptsova, Y. Pakhomov, V. Kosenko // International Journal of Engineering and Technology (UAE, ISSN: 2227-524X). – 2018. – № 2.23. – С. 73–76 (Indexed by Scopus, IEEE Xplore).

18. Пахомов Ю. В. Методы автоматизации проектирования легкотестируемых компьютерных систем и устройств на основе цифровых автоматов / М. А. Мирошник, Л. А. Клименко, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. – 2018. – №4 (131). – С. 3–10 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, Elibrary.ru).

19. Pakhomov Y. Methods for Designing Self-Checking Digital Machines / [M. Miroshnyk, E. Kulak, Y. Pakhomov et al.] // Telecommunications and Radio Engineering, USA. – 2017. – № 15. – С. 1367– 1377 (Indexed by Scopus, IEEE Xplore).

20. Pakhomov Y. Konstruktion of Distributed Information Management Systems for Accounting and Control of Energy Consumption at the Example of Gas / V. Kotukh, Y. Pakhomov // Telecommunications and Radio Engineering, USA. – 2016. – № 7. – С. 631– 641 (Indexed by Scopus, IEEE Xplore).

21. Пахомов Ю. В. Разработка алгоритма прогнозирования функционально-технического состояния изделий газового оборудования и трубопроводных систем / Ю. В. Пахомов // Вестник БГТУ им. В. Г. Шухова: науч.-теор. журнал. – 2015. – № 3. – С. 90–97 (Реферується або індексується міжнародними наукометричними базами, Elibrary.ru, РІНЦ, ІАС, Science index).

22. Пахомов Ю. В. Основные принципы создания единой распределенной системы автоматического контроля и учета энергоресурсов на примере газовой отрасли / В. Г. Котух, Ю. В. Пахомов // Інформаційно-керуючі системи на залізничному транспорті: науково-техн. журнал. – Харків: УкрДУЗТ, 2015. – № 3. – С. 48–55 (Реферується або індексується міжнародними наукометричними базами Google Scholar, РІНЦ, http://elibrary.ru/title_about.asp?id=33934).

23. Пахомов Ю. В. Построение распределенных информационно-управляющих систем учета и контроля энергоресурсов на примере газовой отрасли / В. Г. Котух, Ю. В. Пахомов // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харків: ХНУРЕ, 2015. – № 182.– С. 65–72 (Реферується або індексується міжнародними наукометричними базами Google Scholar, НБУВ, eLIBRARY.RU).

24. Пахомов Ю. В. Исследование распределения тепловых потоков при импульсной лазерной сварке корпусов датчиков для газового оборудования и трубопроводных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов //

Вестник БГТУ им. В. Г. Шухова. – 2014. – № 5. – С. 96–101 (Реферується або індексується міжнародними базами НЕБ Elibrary.ru, РІНЦ, ІАС, Science index).

25. Пахомов Ю. В. Технологическая концепция создания АСУ ТП для объектов энергоснабжения на базе цифровых распределенных систем / М. А. Мирошник, В. Г. Котух, Ю. В. Пахомов // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харків: ХНУРЕ, 2014. – № 179. – С. 131–137 (Реферується або індексується міжнародними базами Google Scholar, НБУВ, Elibrary.ru).

26. Пахомов Ю. В. Методы расчета упругих элементов мембранно-балочного типа в датчиках для газового оборудования и трубопроводных систем / Ю. В. Пахомов // Комунальне господарство міст: наук.-техн. зб. – Харків: ХНУМГ ім. О. М. Бекетова, 2014. – № 116. – С. 64–66 (Реферується або індексується міжнародними наукометричними базами Google Scholar, WorldCat, DOAJ, НБУВ).

ДОДАТОК Б

ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ

1. IEEE East-West Design & Test Symposium (EWDTS'2018, Kazan, Russia, 14-17 Sept. 2018) – 6 p.
2. 28-th International Scientific Symposium "Metrology and Metrology Assurance" (Sozopol, Bulgaria, 10–14 Sept. 2018). – P. 60-64 (Очна участь).
3. Інформаційно-керуючі системи на залізничному транспорті: 31-а міжнародна науково-практична конференція. (Харків, УкрДУЗТ, 24–26 жовт. 2018р., очна участь).
4. IEEE East-West Design & Test Symposium (EWDTS'2017, Novi Sad, Serbia, Sept. 27–Oct. 2, 2017). – P 203-208 (Очна участь).
5. Проблеми інформатики та моделювання (ПІМ-2017): 17-та міжнародна науково-технічна конференція (Харків, НТУ «ХПІ», 11–15 вер. 2017р., очна участь).
6. Інформаційно-керуючі системи на залізничному транспорті: 30-а міжнародна науково-практична конференція (Харків, УкрДУЗТ, 26–27 жовт. 2017р., очна участь).
7. Інформаційні технології та комп'ютерна інженерія: 5-та міжнародна науково-практична конференція (Івано-Франківськ, Вінниця, 27–29 трав. 2015р., очна участь).
8. Інформаційні технології та комп'ютерна інженерія: 4-та міжнар. наук.-практ. конф. (Вінниця, 28–30 трав. 2014р., очна участь).
9. Інформаційно-керуючі системи на залізничному транспорті: 27-а міжнар. наук.-практ. конф. (Харків, УкрДУЗТ, 24–26 вер. 2014р., очна участь).

ДОДАТОК В

ДОКУМЕНТИ, ЩО ПІДТВЕРДЖУЮТЬ ВПРОВАДЖЕННЯ



АКТ

про впровадження в навчальний процес ХНУРЕ результатів дисертаційної роботи Пахомова Ю.В. «Моделі та методи тестопридатного проектування критичних систем логічного управління на основі кінцевих автоматів» за спеціальністю 05.13.05 - "Комп'ютерні системи та компоненти".

Комісія у складі: зав.каф. АПОТ проф. Чумаченко С.В., доц. каф. АПОТ Шкіля О.С., проф. каф. АПОТ Литвинової Є.І. розглянула матеріали дисертаційної роботи Пахомова Ю.В., які використовуються в навчальному процесі кафедри АПОТ ХНУРЕ, і прийшла до наступного висновку.

Розроблені у дисертаційній роботі методи пошуку помилок проектування у HDL-моделях цифрових автоматів дозволяють проводити неруйнівний діагностичний експеримент навіть для автоматів з несправностями функцій переході, що суттєво підвищує глибину пошуку помилок проектування. Метод розширення вхідного алфавіту кінцевого автомата за рахунок введення додаткового стовпця в таблицю переходів-виходів автомата суттєво підвищили тестопридатність моделей цифрових автоматів за рахунок можливості встановлювання автомату в довільний стан.


У навчальному процесі кафедри АПОТ ХНУРЕ результати дисертаційної роботи Пахомова Ю.В. використовуються у таких навчальних дисциплінах.


1. У навчальній дисципліні «Логічне моделювання» для бакалаврів напрямку «Комп'ютерна інженерія» у лекційному матеріалі по темі «Побудова мовних моделей кінцевих автоматів». та у лабораторних роботах по темі «Верифікація мовних моделей кінцевих автоматів».

2. У навчальній дисципліні для бакалаврів напрямку «Комп'ютерна інженерія» «Основи комп'ютерної діагностики» у лекційному матеріалі та практичних заняттях по темі «Методи пошуку дефектів у цифрових пристроях».

3. У навчальній дисципліні «Теорія проектування СКС та мереж» для магістрів спеціальності «Комп'ютерна інженерія» спеціалізації «Спеціалізовані комп'ютерні системи» у лекційному матеріалі по темі «Системи автоматизованого проектування цифрових пристроїв».

 Зав.каф. АПОТ проф. Чумаченко С.В.,

 Доц. каф. АПОТ Шкіль О.С.,

 Проф. каф. АПОТ Литвинова Є.І.

Акціонерне товариство
«Укртрансгаз»
Філія «Науково-дослідний
інститут транспорту газу»



Ukrtransgaz
Joint Stock Company
Branch «R&D Institute
of Gas Transportation»

Україна, 61004, м. Харків,
вул. Конєва, 16
тел.: +38 (057) 733-31-85
e-mail: i-office@utg.ua



ISO 50001
ISO 9001
ISO 14001
OHSAS 18001

16, Koneva st.
Kharkov, 61004, Ukraine
tel.: +38 (057) 733-31-85
e-mail: i-office@utg.ua

в.д.г. 25.10.2018, № 687-01/БК

На № _____

Довідка

про впровадження результатів дисертаційної роботи

Пахомова Юрія Васильовича

на тему «Моделі та методи тестопридатного проектування критичних систем логічного управління на основі кінцевих автоматів»

за спеціальністю 05.13.05 – «Комп'ютерні системи та компоненти»

На сьогодні в газопостачанні відбуваються суттєві зміни, пов'язані із впровадженням нових технологій автоматизації та ресурсозбереження. У системах енерго- та газопостачання широко застосовуються спеціальні локальні системи управління і регулювання, розташовані на віддаленій місцевості, наприклад на трансформаторних підстанціях, газорозподільних вузлах, пунктах обліку енергоспоживання тощо.

У роботі розроблено методики апаратного діагностування локальних пристроїв управління та регулювання технологічно відокремленими об'єктами електроенергетики і газопостачання. Алгоритми управління описуються граф-схемою алгоритму, яка подана

у вигляді моделі графу переходів кінцевого автомата. Для реалізації алгоритму діагностування використовується спосіб обходу усіх дуг графа. Модель автомата управління і пристрою діагностування описуються на мові опису апаратури VHDL з наступним синтезом та використанням пристроїв програмованої логіки. Проаналізовано необхідну кількість апаратного забезпечення для різних варіантів реалізації пристрою діагностування.

Результати дисертаційної роботи Пахомова Ю.В. були застосовані в діяльності підрозділів Науково-дослідного інституту транспорту газу АТ «Укртрансгаз» для створення систем контролю автоматичних газорозподільних станцій, оснащених сучасним автоматичним обладнанням на базі цифрової техніки. Результати дисертаційної роботи дозволяють суттєво скоротити час виявлення та усунення можливих дефектів,

які виникають на об'єктах газотранспортної системи, та підвищити ступінь надійності транспортування газу.

Заступник директора з наукової роботи

Науково-дослідного інституту транспорту газу АТ «Укртрансгаз», к.т.н., доцент



Пономарьов Ю. В.



РЕГІОНАЛЬНА
ГАЗОВА КОМПАНІЯ

ХАРКІВМІСЬКГАЗ

від 15.11.2018р

№935-02/Д

Довідка

про впровадження результатів дисертаційної роботи

Пахомова Юрія Васильовича

на тему *«Моделі та методи тестопридатного проектування критичних систем логічного управління на основі кінцевих автоматів»*

за спеціальністю 05.13.05 – «Комп'ютерні системи та компоненти»

В електроенергетиці і газопостачанні досить поширені віддалені від населених пунктів об'єкти управління, які працюють без участі людини. До таких об'єктів відносяться трансформаторні підстанції, газорозподільні станції, пункти обліку енергоспоживання тощо. Для забезпечення надійної та безпечної роботи зазначених об'єктів необхідно своєчасне проведення регламентних робіт з їх технічного обслуговування та діагностування.

У роботі було проведено теоретичні дослідження зі створення цифрових автоматних пристроїв логічного управління, для яких, зазвичай, не є критичними швидкодія та технологічна платформа реалізації. Виходячи з цього, зазначені цифрові автоматні пристрої логічного управління реалізуються на технологічній базі мікроконтролерних систем або на базі програмованих логічних інтегральних схем (ПЛІС). Розроблені автоматні моделі апаратної системи діагностування цифрових пристроїв управління в електроенергетиці і газопостачанні. Ці моделі представлені на мовах опису апаратури для виконання їх верифікації. Схемна реалізація керуючого пристрою та пристрою діагностування виконується за допомогою засобів автоматизованого проектування.

Результати дисертаційної роботи Пахомова Ю.В. були застосовані в діяльності підрозділів Приватного акціонерного товариства «Харківміськгаз» для створення надійних систем контролю у міських автоматизованих газорозподільчих пунктах, які оснащені сучасним автоматичним обладнанням на базі цифрової техніки. Результати дисертаційної роботи дозволяють суттєво скоротити час виявлення та усунення можливих дефектів, які виникають на об'єктах газотранспортної системи, та підвищити ступінь надійності транспортування газу.

Директор технічний

Я.О. Шувалова



ПРАТ «Енергооблік», Україна, 61052, м. Харків-52, а/с 332
 ☎ +38 (057) 734-98-51/52/53; факс +38 (057) 734-99-16
 ✉ sales@energo.kh.ua  http://www.energo.kh.ua

№ 337/ВК від 28.11.2018р.

Довідка

про впровадження результатів дисертаційної роботи

Пахомова Юрія Васильовича

на тему *«Моделі та методи тестопридатного проектування критичних систем логічного управління на основі кінцевих автоматів»*
 за спеціальністю 05.13.05 – «Комп'ютерні системи та компоненти»

У дисертаційній роботі було проведено теоретичні дослідження зі створення цифрових автоматних пристроїв логічного управління та апаратного діагностування, для яких, зазвичай, не є критичними швидкодія та технологічна платформа реалізації. Крім цього, вони мають бути тестопридатними і простими в налаштуванні.

У роботі запропоновано новий підхід при автоматизованому проектуванні тестопридатних кінцевих автоматів за рахунок побудови моделей на мовах опису апаратури (HDL-моделей), у яких додаткові переходи реалізуються шляхом включення умовних операторів в опис функцій переходів в автоматний шаблон на мові опису апаратури. Розроблені HDL-моделі моделюються та синтезуються інструментальними засобами систем автоматизованого проектування цифрових пристроїв на технологічній платформі ПЛІС. Розроблений програмний модуль з візуальним інтерфейсом для вводу графу переходів керуючого автомату управління для автоматизованої побудови тестопридатних HDL-моделей автоматів у формі автоматного шаблону у синтезованій підмножині мови опису апаратури VHDL.

Результати дисертаційної роботи Пахомова Юрія Васильовича були впроваджені у виробничій діяльності ПрАТ «Енергооблік», яке є розробником мікропроцесорних вимірювальних приладів і систем, під час розробки систем тестопридатного проектування вимірювальних приладів. Прилади виробництва ПрАТ «Енергооблік» успішно працюють на підприємствах України і країн СНД, в тому числі на нафтохімічних виробництвах та магістральних газопроводах.

Директор ПрАТ «Енергооблік», к.т.н.

Стеценко А. А.



ДОДАТОК Г

VHDL-МОДЕЛЬ КЕРУЮЧОГО АВТОМАТА КП ГРП СИСТЕМА ВЕРИФІКАЦІЇ ТА ДІАГНОСТУВАННЯ

Г.1 VHDL-модель керуючого автомата КП ГРП

```

-- Finite state machine that generate output signals for
-- control object to execute data operations

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity GAS_FSM is
    port (
        clk: in STD_LOGIC;
        reset: in STD_LOGIC;
        X: in STD_LOGIC_VECTOR (0 to 7);
        Y: out STD_LOGIC_VECTOR (1 to 14));
end GAS_FSM;

architecture GAS_FSM_arch of GAS_FSM is

    -- Not good use keyword "signal" because states always
    -- must be initialized!!!
    constant A1: std_logic_vector(3 downto 0) := "0001";
    constant A2: std_logic_vector(3 downto 0) := "0010";
    constant A3: std_logic_vector(3 downto 0) := "0011";
    constant A4: std_logic_vector(3 downto 0) := "0100";
    constant A5: std_logic_vector(3 downto 0) := "0101";
    constant A6: std_logic_vector(3 downto 0) := "0110";
    constant A7: std_logic_vector(3 downto 0) := "0111";
    constant A8: std_logic_vector(3 downto 0) := "1000";
    constant A9: std_logic_vector(3 downto 0) := "1001";
    signal State, NextState: std_logic_vector (3 downto 0);

begin
    State_CurrentState: process (clk, reset)
    begin
        if rising_edge(clk) then
            if reset='1' then State <= A1;
            else State <= NextState;
            end if;
        end if;
    end process;
end GAS_FSM_arch;

```

```

State_NextState: process (X, State)
begin
  Y<= (others=>'0');
  case State is
  -- X(0) is a bit that is marked on algorithm graph as Z
    when "0001" =>
      if (X(0)='1') then
        NextState <= A2;
        Y(1) <= '1';
      elsif (X(0) = '1') then
        NextState <= A1;
      else NextState <= A1;
      end if;

    when "0010" =>
      NextState <= A3;
      Y(2) <= '1';

    when "0011" =>
      if (X(1)='1') then
        NextState <= A4;
        Y(3) <= '1';
      elsif (X(1) = '0') then
        NextState <= A1;
        Y(8) <= '1';
      else NextState <= A1;
      end if;

    when "0100" =>
      if (X(2)='1') then
        NextState <= A5;
        Y(4) <= '1';
      elsif (X(2) = '0') then
        NextState <= A1;
        Y(9) <= '1';
      else NextState <= A1;
      end if;

    when "0101" =>
      if (X(3)='1') then
        NextState <= A6;
        Y(5) <= '1';
      elsif (X(3)='0') then
        NextState <= A5;
        Y(12) <= '1';
      else NextState <= A1;
      end if;

    when "0110" =>
      if (X(4)='1') then
        NextState <= A7;
        Y(6) <= '1';
      elsif (X(4) = '0') then

```

```

        NextState <= A1;
        Y(10) <= '1';
    else NextState <= A1;
    end if;

    when "0111" =>
        if (X(5)='1') then
            NextState <= A8;
            Y(7) <= '1';
        elsif (X(5)='0') then
            NextState <= A9;
            Y(13) <= '1';
        else NextState <= A1;
        end if;

    when "1000" =>
        if (X(6)='0') then
            NextState <= A1;
            Y(11) <= '1';
        elsif (X(6) = '1') then
            NextState <= A1;
        else NextState <= A1;
        end if;

    when "1001" =>
        if (X(7)='1') then
            NextState <= A8;
            Y(7) <= '1';
        elsif (X(7) = '1') then
            NextState <= A1;
            Y(14) <= '1';
        else NextState <= A1;
        end if;
    -- No null for synthesis!!!
    when others => NextState <= A1;

    end case;

end process;

end GAS_FSM_arch;

```

Г.2 Верифікація VHDL-моделі керуючого автомата КП ГРП

```

library ieee;
use ieee.STD_LOGIC_UNSIGNED.all;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

-- Add your library and packages declaration here ...

```

```

entity gas_fsm_tb is
end gas_fsm_tb;

architecture TB_ARCHITECTURE of gas_fsm_tb is
  -- Component declaration of the tested unit
  component gas_fsm
  port(
    clk : in STD_LOGIC;
    reset : in STD_LOGIC;
    X : in STD_LOGIC_VECTOR(0 to 7);
    Z : in STD_LOGIC;
    Y : out STD_LOGIC_VECTOR(0 to 13) );
  end component;

  -- Stimulus signals - signals mapped to the input and inout
  ports of tested entity
  signal clk : STD_LOGIC;
  signal reset : STD_LOGIC;
  signal X : STD_LOGIC_VECTOR(0 to 7);
  signal Z : STD_LOGIC;
  -- Observed signals - signals mapped to the output ports of
  tested entity
  signal Y : STD_LOGIC_VECTOR(0 to 13);

  -- Add your code here ...

begin

  -- Unit Under Test port map
  UUT : gas_fsm
    port map (
      clk => clk,
      reset => reset,
      X => X,
      Z => Z,
      Y => Y
    );

  clk_process: process
  begin

    CLK <= '0';
    wait for 50 ns;

    CLK<='1';
    wait for 50 ns;

    wait for 0 ns;

  end process;

  reset_process: process
  begin

```



```

        RESET <= '1';
        wait for 10 ns;

        RESET <= '0';
        wait for 1ms;

    end process;

    x_process: process
    begin

        Z<='0'; X<="00000000";
        wait for 10 ns;

        Z<='1'; X<="11000010";
        wait for 1ms;

    end process;

end TB_ARCHITECTURE;

configuration TESTBENCH_FOR_gas_fsm of gas_fsm_tb is
    for TB_ARCHITECTURE
        for UUT : gas_fsm
            use entity work.gas_fsm(gas_fsm_arch);
        end for;
    end for;
end TESTBENCH_FOR_gas_fsm;

```

Для верифікації розглянемо послідовність переходів A1-A2-A3-A4-A5-A6-A1. Перехід зі стану в стан відповідає за ініціацію блоків і датчиків, описаних в лістингу Г.1. Відбувається ініціація автомата (запуск різних блоків для початку роботи УА (A1 - A2; $X(0) = 1 \Rightarrow Y1 = 1$)) опитування датчика тиску (A3-A4; $Y2 = 1$), температури (A4-A5; $X(1) = 1 \Rightarrow Y3 = 1$) та газоконденсату (A5-A6; $X(3) = 1 \Rightarrow Y4 = 1$). Після опитування кожного датчика відбувається порівняння їх показників з необхідними для подальшої роботи. Позитивна перевірка ($X(1) = X(2) = X(3) = 1$) показує, що показники не перевищують допустимі і автомат далі продовжує коректну роботу. Перевірка наступного датчика мінімального рівня газоконденсату не проходить (A6-A7; $X(4) = 0$) і виникає аварійна ситуація ($Y(10) = 1$), яка повертає автомат в початковий стан (A1). Результати верифікації представлені на рис. Г.1.

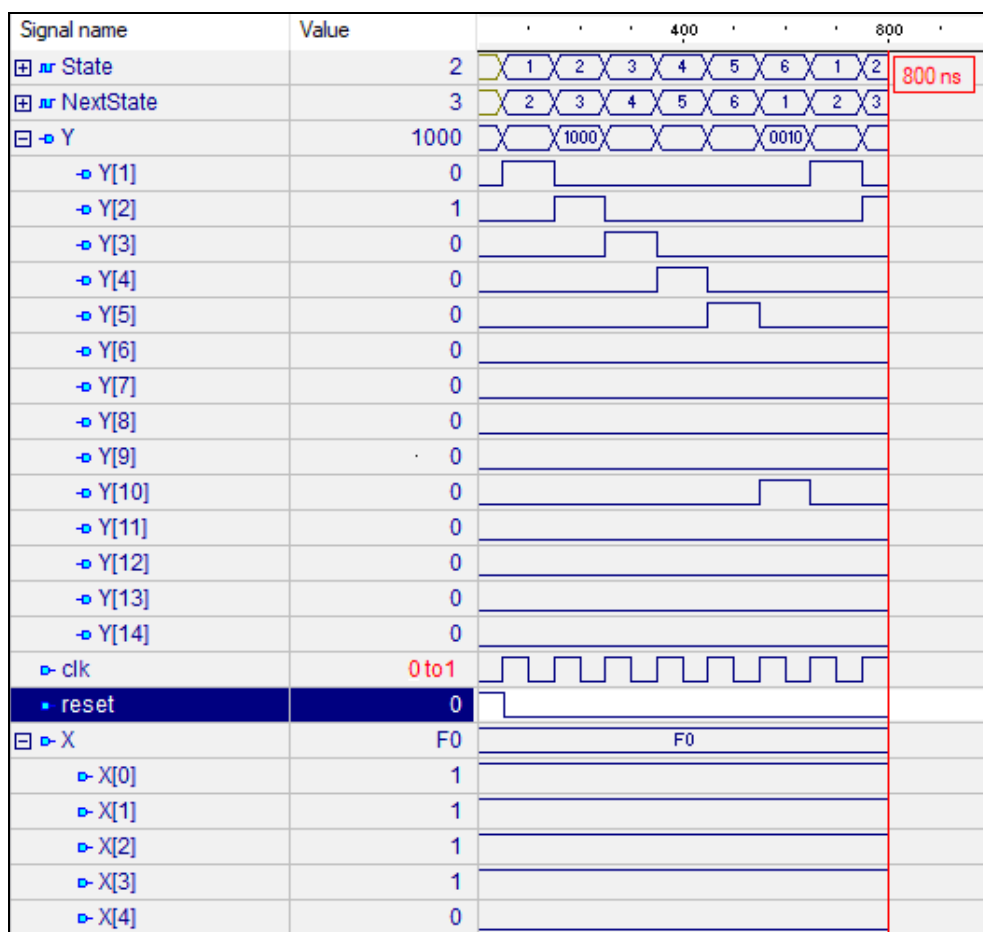
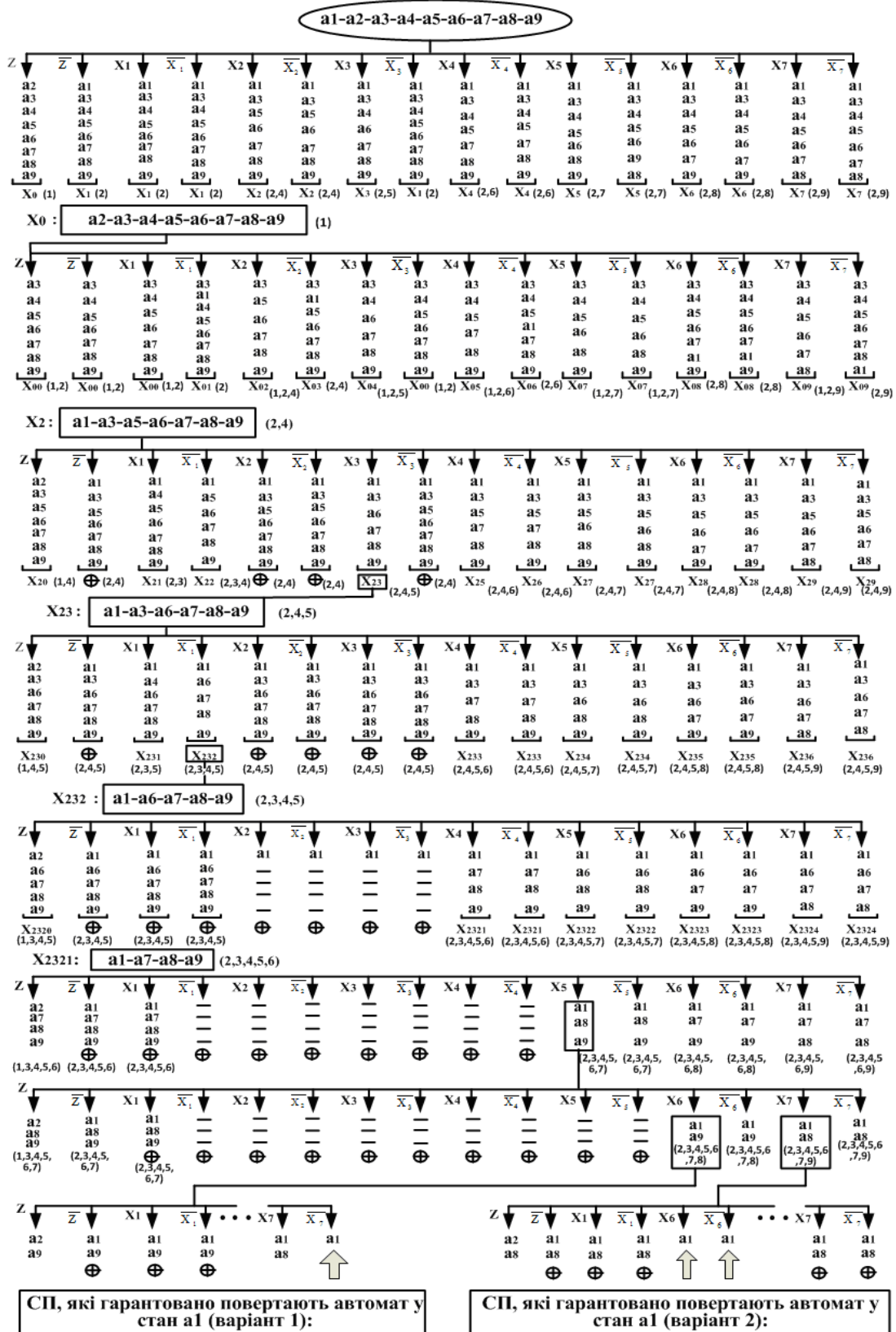


Рисунок Г.1 – Результат моделювання (waveform) роботи КА КП ГРП

ДОДАТОК Д

МОДЕЛЬ ПОБУДОВИ СИНХРОНІЗУЮЧИХ ПОСЛІДОВНОСТЕЙ ДЛЯ КА КП ГРП

Рисунок Д.1 – Повне синхронізує дерево КП ГРП



Таблиця Д.1 – Синхронізуючі послідовності.

№ з/п	Варіант 1	№ з/п	Варіант 2
1	$X_2; X_3; \overline{X_1}; X_4; X_5; X_6; \overline{X_7}$.	1	$X_2; X_3; \overline{X_1}; X_4; X_5; X_7; X_6$.
2	$X_2; X_3; \overline{X_1}; X_4; X_5; \overline{X_6}; \overline{X_7}$.	2	$X_2; X_3; \overline{X_1}; X_4; X_5; X_7; \overline{X_6}$.
3	$X_2; X_3; \overline{X_1}; X_4; \overline{X_5}; \overline{X_6}; \overline{X_7}$.	3	$X_2; X_3; \overline{X_1}; X_4; X_5; \overline{X_7}; \overline{X_6}$.
4	$X_2; X_3; \overline{X_1}; X_4; \overline{X_5}; \overline{X_6}; \overline{X_7}$.	4	$X_2; X_3; \overline{X_1}; X_4; \overline{X_5}; \overline{X_7}; \overline{X_6}$.
5	$X_2; X_3; \overline{X_1}; X_4; \overline{X_5}; X_6; \overline{X_7}$.	5	$X_2; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; X_6$.
6	$X_2; X_3; \overline{X_1}; \overline{X_4}; X_5; X_6; \overline{X_7}$.	6	$X_2; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; X_6$.
7	$X_2; X_3; \overline{X_1}; X_4; \overline{X_5}; X_6; \overline{X_7}$.	7	$X_2; X_3; \overline{X_1}; \overline{X_4}; \overline{X_5}; X_7; X_6$.
8	$X_2; X_3; \overline{X_1}; X_4; X_5; \overline{X_6}; \overline{X_7}$.	8	$X_2; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; X_6$.
9	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; X_6; \overline{X_7}$.	9	$X_2; X_3; \overline{X_1}; X_4; \overline{X_5}; \overline{X_7}; X_6$.
10	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; \overline{X_6}; \overline{X_7}$.	10	$X_2; X_3; \overline{X_1}; X_4; \overline{X_5}; X_7; X_6$.
11	$\overline{X_2}; X_3; \overline{X_1}; X_4; \overline{X_5}; \overline{X_6}; \overline{X_7}$.	11	$X_2; X_3; \overline{X_1}; X_4; X_5; \overline{X_7}; X_6$.
12	$\overline{X_2}; X_3; \overline{X_1}; X_4; \overline{X_5}; \overline{X_6}; \overline{X_7}$.	12	$X_2; X_3; \overline{X_1}; X_4; \overline{X_5}; X_7; \overline{X_6}$.
13	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; \overline{X_6}; \overline{X_7}$.	13	$X_2; X_3; \overline{X_1}; \overline{X_4}; X_5; \overline{X_7}; X_6$.
14	$\overline{X_2}; X_3; \overline{X_1}; X_4; \overline{X_5}; X_6; \overline{X_7}$.	14	$X_2; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; \overline{X_6}$.
15	$\overline{X_2}; X_3; \overline{X_1}; X_4; \overline{X_5}; X_6; \overline{X_7}$.	15	$X_2; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; \overline{X_6}$.
16	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; X_6; \overline{X_7}$.	16	$X_2; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; X_6$.
17		17	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; X_7; X_6$.
18		18	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; X_7; \overline{X_6}$.
19		19	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; \overline{X_7}; \overline{X_6}$.
20		20	$\overline{X_2}; X_3; \overline{X_1}; X_4; \overline{X_5}; X_7; \overline{X_6}$.
21		21	$\overline{X_2}; X_3; \overline{X_1}; \overline{X_4}; \overline{X_5}; X_7; \overline{X_6}$.
22		22	$\overline{X_2}; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; X_6$.
23		23	$\overline{X_2}; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; X_6$.
24		24	$\overline{X_2}; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; X_6$.
25		25	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; X_7; X_6$.
26		26	$\overline{X_2}; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; \overline{X_6}$.
27		27	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; X_7; X_6$.
28		28	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; X_7; X_6$.
29		29	$\overline{X_2}; X_3; \overline{X_1}; X_4; X_5; X_7; X_6$.
30		30	$\overline{X_2}; X_3; \overline{X_1}; X_4; \overline{X_5}; X_7; \overline{X_6}$.
31		31	$\overline{X_2}; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; \overline{X_6}$.
32		32	$\overline{X_2}; X_3; \overline{X_1}; \overline{X_4}; X_5; X_7; \overline{X_6}$.

ДОДАТОК Е

VHDL-МОДЕЛІ АПАРАТНОГО ПРИСТРОЮ ДІАГНОСТУВАННЯ

Е.1 HDL-модель пристрою діагностування з кодуванням вхідного масиву за варіантом 1.

Лістинг Е.1 – VHDL-модель ПД за варіантом 1

```
--The diagnostic device that generates a set of conditions for the
operation of the control device
--a set of conditions (8-bit register X) is a check of the sequence of
transitions
--on which the control automaton must pass to check for correctness
working

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity GAS_DD_FSM is
    port (
        clk: in STD_LOGIC;
        reset: in STD_LOGIC;
        D: in STD_LOGIC_VECTOR (1 to 10);
        Yx: out STD_LOGIC_VECTOR (0 to 7));
end GAS_DD_FSM;

architecture arch of GAS_DD_FSM is

    signal State, NextState: std_logic_vector(3 downto 0);
    -- Not good use keyword "signal" because states always
    -- must be initialized!!!
    constant A1: std_logic_vector(3 downto 0) := "0001";
    constant A2: std_logic_vector(3 downto 0) := "0010";
    constant A3: std_logic_vector(3 downto 0) := "0011";
    constant A4: std_logic_vector(3 downto 0) := "0100";
    constant A5: std_logic_vector(3 downto 0) := "0101";
    constant A6: std_logic_vector(3 downto 0) := "0110";
    constant A7: std_logic_vector(3 downto 0) := "0111";
    constant A8: std_logic_vector(3 downto 0) := "1000";
    constant A9: std_logic_vector(3 downto 0) := "1001";
    -- Signal for copying input sequence from external memory
    signal DD: std_logic_vector (1 to 10);

begin
    State_CurrentState: process (clk,reset)
    begin
        if rising_edge(clk) then
            if reset='1' then
```

```

        -- Copying input sequence from external memory
        DD <= D;
        State <= A1;
    else State <= NextState;
    end if;
end if;
end process;

-- Sensitivity list must include control signals
-- (State for case enumeration and input register
-- for transit condition
State_NextState: process (State,DD)
begin
    Yx<= (others=>'0');
    case State is

        when "0001" =>
            if(DD(2) = '1') then
                NextState <= A2;
                Yx(0) <= '1';
            elsif (DD(2) = '0') then
                NextState <= A1;
            else NextState <= A1;
            end if;

        when "0010" =>
            NextState <= A3;

        when "0011" =>
            if(DD(4) = '1') then
                NextState <= A4;
                Yx(1) <= '1';
            elsif(DD(4) = '0') then
                NextState <= A1;
            else NextState <= A1;
            end if;

        when "0100" =>
            if(DD(5) = '1') then
                NextState <= A5;
                Yx(2) <= '1';
            elsif (DD(5) = '0') then
                NextState <= A1;
            else NextState <= A1;
            end if;

        when "0101" =>
            if(DD(6) = '1') then
                NextState <= A6;
                Yx(3) <= '1';
            elsif (DD(6) = '0') then
                NextState <= A5;
            else NextState <= A1;
            end if;

        when "0110" =>
            if(DD(7) = '1') then
                NextState <= A7;

```

```

        Yx(4) <= '1';
    elsif (DD(7) = '0') then
        NextState <= A1;
    else NextState <= A1;
    end if;

-- WARNING! we can transit from A7 only to A8 or A9
when "0111" =>
    if(DD(8) = '1') then
        NextState <= A8;
        Yx(5) <= '1';
    elsif (DD(8) = '0') then
        NextState <= A9;
    else NextState <= A1;
    end if;

--WARNING! problem with a multiple transition from A8 to A1
--when "1000" =>
--if(D(9) = "0001") then

when "1001" =>
    if(DD(9) = '1') then
        NextState <= A8;
        Yx(7) <= '1';
    elsif(DD(9) = '0') then
        NextState <= A1;
    else NextState <= A1;
    end if;
    -- No null for synthesis machine template!!!
    when others => NextState <= A1;
end case;

end process;

end arch;
```

В лістингу E.2 представлений TestBench введення вхідної послідовності станів a1 – a2 – a3 – a4 – a5 – a6 – a1 за першим варіантом кодування, а на рис. E.1 – часова діаграма (Waveform) моделювання даної послідовності в середовищі Active-HDL.

E.2 – Фрагмент TestBench для введення вхідної послідовності за варіантом 1

```

-- Unit Under Test port map
UUT : gas_dd_fsm
port map (
    clk => clk,
    reset => reset,
    D => D,
```

```

        Yx => Yx
    );

    clk_process: process
    begin
        clk <= '0';
        wait for 50 ns;
        clk<='1';
        wait for 50 ns;
        wait for 0 ns;

    end process;

    some_process:process
    begin
        reset<='1';

        -- Each bit of input array characterizes not crash
        transition ('1') between state and crash transition ('0')

        D<="1111110000";

        wait for 60 ns;
        reset <= '0';
        wait;
    end process;...

```

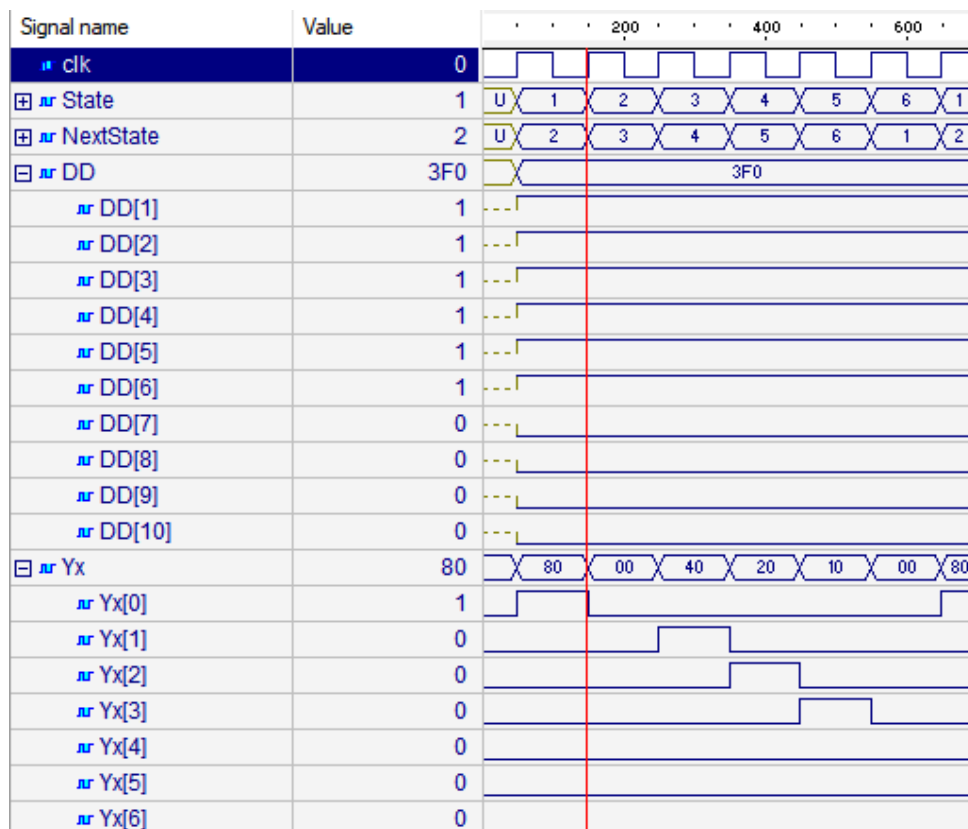


Рисунок Е.1 – Waveform роботи КП з бітовим масивом, з масивом, що описує послідовність (A1-A2-A3-A4-A5-A6-A1)

Лістинг Е.3 – VHDL-модель ПД за варіантом 2

```

--The diagnostic device that generates a set of conditions for the
operation of the control device
--a set of conditions (8-bit register X) is a check of the sequence of
transitions
--on which the control automaton must pass to check for correctness
working
-- The feature of device is concrete state that can not be changed
-- (this code can optimized using ROM)

library IEEE;
use IEEE.std_logic_1164.all;
--Package for creating array of states
--Each state is signal of std_logic_vector
--(state is an item of array)
PACKAGE heap_arr_pkg IS
    type TestSequence is array (1 to 10) of std_logic_vector (1 to
4);
END;

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

--Use package for creating array of states (array of
std_logic_vector)
USE work.heap_arr_pkg.all;

entity GAS_DD_Listing3 is
    port (
        clk: in STD_LOGIC;
        reset: in STD_LOGIC;
        D: in TestSequence;
        Yx: out STD_LOGIC_VECTOR (0 to 7));
end GAS_DD_Listing3;

architecture arch of GAS_DD_Listing3 is

    signal State, NextState: std_logic_vector(3 downto 0);
    -- Not good use keyword "signal" because states always
    -- must be initialized!!!
    constant A1: std_logic_vector(3 downto 0) := "0001";
    constant A2: std_logic_vector(3 downto 0) := "0010";
    constant A3: std_logic_vector(3 downto 0) := "0011";
    constant A4: std_logic_vector(3 downto 0) := "0100";
    constant A5: std_logic_vector(3 downto 0) := "0101";
    constant A6: std_logic_vector(3 downto 0) := "0110";
    constant A7: std_logic_vector(3 downto 0) := "0111";
    constant A8: std_logic_vector(3 downto 0) := "1000";
    constant A9: std_logic_vector(3 downto 0) := "1001";
    -- Signal for copying input sequence from external memory
    signal DD: TestSequence;

begin

```

```

State_CurrentState: process (clk,reset)
begin
    if rising_edge(clk) then
        if reset='1' then
            -- Copying input sequence from external memory
            DD <= D;
            State <= A1;
        else State <= NextState;
        end if;
    end if;
end process;

-- Sensitivity list must include control signals
-- (State for case enumeration and input register
-- for transit condition)
State_NextState: process (State, DD)
begin
    Yx<= (others=>'0');
    case State is

        when "0001" =>
            if(DD(2) = "0010") then
                NextState <= A2;
                Yx(0) <= '1';
            else NextState <= A1;
            end if;

        when "0010" =>
            NextState <= A3;

        when "0011" =>
            if(DD(4) = "0100") then
                NextState <= A4;
                Yx(1) <= '1';
            else NextState <= A1;
            end if;

        when "0100" =>
            if(DD(5) = "0101") then
                NextState <= A5;
                Yx(2) <= '1';
            else NextState <= A1;
            end if;

        when "0101" =>
            if(DD(6) = "0110") then
                NextState <= A6;
                Yx(3) <= '1';
            elsif (DD(6) = "0101") then
                NextState <= A5;
            else NextState <= A1;
            end if;

        when "0110" =>
            if(DD(7) = "0111") then
                NextState <= A7;
                Yx(4) <= '1';
            else NextState <= A1;
    end case;
end process;

```

```

        end if;

        -- WARNING! we can transit from A7 only to A8 or A9
        when "0111" =>
            if(DD(8) = "1000") then
                NextState <= A8;
                Yx(5) <= '1';
            elsif (DD(8) = "1001") then
                NextState <= A9;
            else NextState <= A1;
            end if;

        --WARNING! problem with a multiple transition from A8 to A1
        --when "1000" =>
        --if(D(9) = "0001") then

        when "1001" =>
            if(DD(9) = "1000") then
                NextState <= A8;
                Yx(7) <= '1';
            else NextState <= A1;
            end if;

        -- No null for synthesis machine template!!!
        when others => NextState <= A1;
    end case;
end process;
end arch;

```

Лістинг Е.4 представляє TestBench введення вхідної послідовності станів a1 – a2 – a3 – a4 – a5 – a6 – a1 за другим варіантом кодування, а на рис. 3.8 - часова діаграма (Waveform) моделювання даної послідовності в середовищі Active-HDL.

Лістинг Е.4 – Фрагмент TestBench для введення вхідної послідовності за варіантом 2

```

-- Unit Under Test port map
    UUT : gas_dd_fsm_listing_3
        port map (
            clk => clk,
            reset => reset,
            D => D,
            Yx => Yx
        );
    clk_process: process
    begin
        clk <= '0'; wait for 50 ns;
        clk<='1'; wait for 50 ns;
        wait for 0 ns;
    end process;

```

```

some_process:process
begin
    reset<='1';
-- Each item of input array is 4-bits field
-- Each 4-bits field means current state of test sequence
    D(1)<= "0001";
    D(2)<= "0010";
    D(3)<= "0011";
    D(4)<= "0100";
    D(5)<= "0101";
    D(6)<= "0110";
    -- It means the end of transitions
    -- It returns to initial state
    D(7)<= "0001";
    -- No matter what is in these items, because
    -- current transition endings at D(7) = "0001"
    D(8)<= "0010";
    D(9)<= "0011";
    D(10)<= "0001";

    wait for 60 ns;
    reset <= '0';
    wait;
end process;...

```

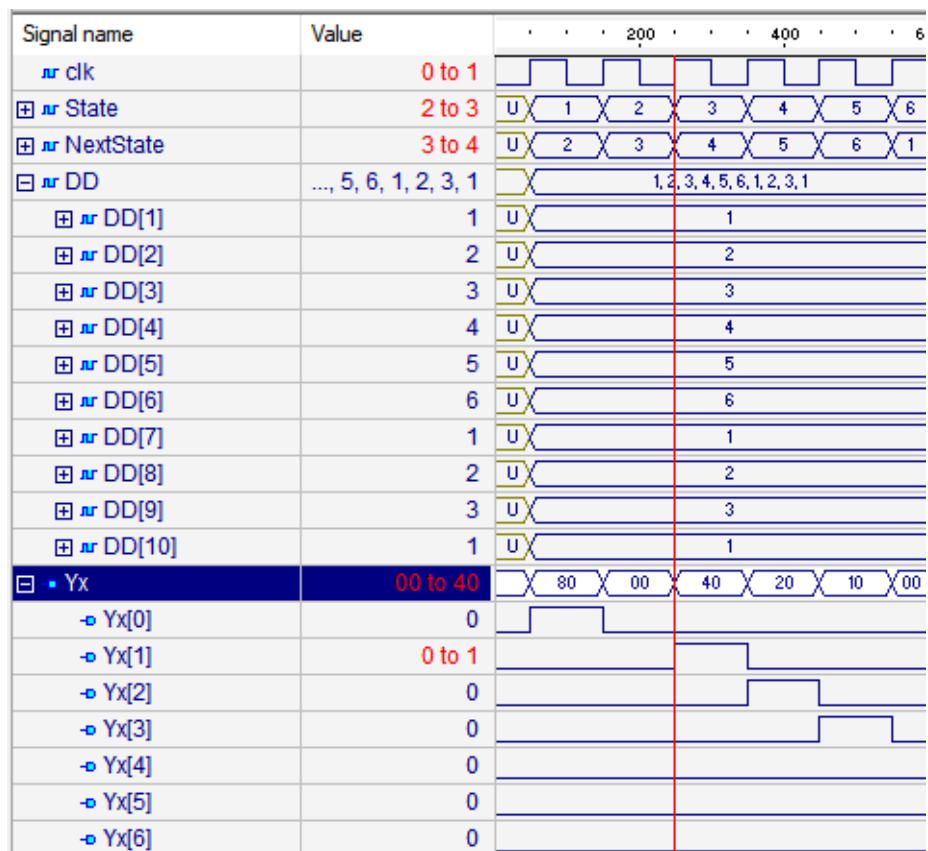


Рисунок Е.2 – Waveform ПД з використанням масива векторів, що містять стан, описаний числом (A1-A2-A3-A4-A5-A6-A1)

Лістинг Е.5 – VHDL-модель ПД за варіантом 3

```

--The diagnostic device that generates a set of conditions for
the operation of the control device
--a set of conditions (8-bit register X) is a check of the
sequence of transitions
--on which the control automaton must pass to check for
correctness    working

library IEEE;
use IEEE.std_logic_1164.all;

--Package for creating array of states
--Each state is signal of std_logic_vector
PACKAGE heap_arr_pkg IS
    type TestSequence is array (1 to 10) of std_logic_vector (1
to 4);
    -- Describe sequence that ROM keeps
    -- For example it's only one sequence
    constant c_Test_sequence: TestSequence :=
("0001", "0010", "0011", "0100", "0101", "0110", "0001", "0001", "0001",
"0001");
END;

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

--Use package for creating array of states (array of
std_logic_vector)
USE work.heap_arr_pkg.all;

entity GAS_Diagnostic_Device is
    port (
        clk: in STD_LOGIC;
        reset: in STD_LOGIC;
        -- No input for reading from ROM
        --D: in TestSequence;
        Yx: out STD_LOGIC_VECTOR (0 to 7));
end GAS_Diagnostic_Device;

architecture arch of GAS_Diagnostic_Device is

    -- If waveform has 'U' in starting point
    --(when reset is 1 and before rising edge)
    -- State and NextState can be initialized there
    signal State: std_logic_vector(1 to 4);
    signal NextState: std_logic_vector(1 to 4);
    --Signal for copying test sequence (reading constant)
    signal DD : TestSequence;

```

```

begin
  --Process for initialization and switching DD array
  State_CurrentState: process (clk,reset,DD)
  begin
    if rising_edge(clk) then
      --ATTENTION! Reset (reset) must load
      --only first sequence in ROM every time
      if (reset='1') then

        --Copying the first sequence from ROM to signal
        DD <= c_Test_sequence;

        -- Synchronize process must not working with signals
        -- that operated in other process (State, NextState)
        -- so NextState has a role DD(2)
        -- Array item needs to be checked if it initial state or not
        elsif(DD(2) = "0001") then

          --Copying from ROM to signal
          --ATTENTION!If we using all sequences there must be shifting
          --an array of all sequences (another sequence except first)
          DD <= c_Test_sequence;

        else
          DD <= DD(2 to 10)&"0001";
        end if;

        --Initiation State and NextState after shifting
        State <= DD(1); NextState <= DD(2);
      end if;
    end process;

  State_NextState: process (State, NextState)
  begin
    Yx<= (others=>'0');
    case State is

      when "0001" =>
        if(NextState = "0010") then
          Yx(0) <= '1';
        end if;

      when "0011" =>
        if(NextState = "0100") then
          Yx(1) <= '1';
        end if;

      when "0100" =>
        if(NextState = "0101") then
          Yx(2) <= '1';
        end if;

      when "0101" =>

```

```

        if(NextState = "0110") then
            Yx(3) <= '1';
        end if;

    when "0110" =>
        if(NextState = "0111") then
            Yx(4) <= '1';
        end if;

    when "0111" =>
        if(NextState = "1000") then
            Yx(5) <= '1';
        end if;

--WARNING! problem with a multiple transition from A8 to A1
--when "1000" =>
--    if(NextState = "0001") then

    when "1001" =>
        if(NextState = "1000") then
            Yx(7) <= '1';
        end if;

        -- In this case null is synthesized
        -- BUT, desirable, use another operator
        -- or operation for avoiding non-synthesize
        when others => null;
    end case;
end process;
end arch;

```

Лістинг 3.6 представляє TestBench введення вхідної послідовності станів a1 – a2 – a3 – a4 – a5 – a6 – a1 за третім варіантом кодування, а на рис. 3.9 – Waveform моделювання даної послідовності в середовищі Active-HDL.

Лістинг Е.6 – Фрагмент TestBench для введення вхідної послідовності за варіантом 3

```

...
-- Unit Under Test port map
    UUT : gas_diagnostic_device
        port map (
            clk => clk,
            reset => reset,
            D => D,
            Yx => Yx
        );

    clk_process: process

```

```

begin

    clk <= '0';
    wait for 50 ns;
    clk<='1';
    wait for 50 ns;
    wait for 0 ns;
end process;

some_process:process

begin
    reset<='1';

-- Each item of input array is 4-bits field
-- Each 4-bits field means state of test sequence
-- First element of array is State and the second is NextState

    D(1)<= "0001";
    D(2)<= "0010";
    D(3)<= "0011";
    D(4)<= "0100";
    D(5)<= "0101";
    D(6)<= "0110";

    -- It means the end of transitions
    -- It returns to initial state

    D(7)<= "0001";

    -- No matter what is in these items, because
    -- current transition endings at D(7) = "0001"
    D(8)<= "0010";
    D(9)<= "0011";
    D(10)<= "0001";

    wait for 60 ns;
    reset <= '0';
    wait;

end process;

```

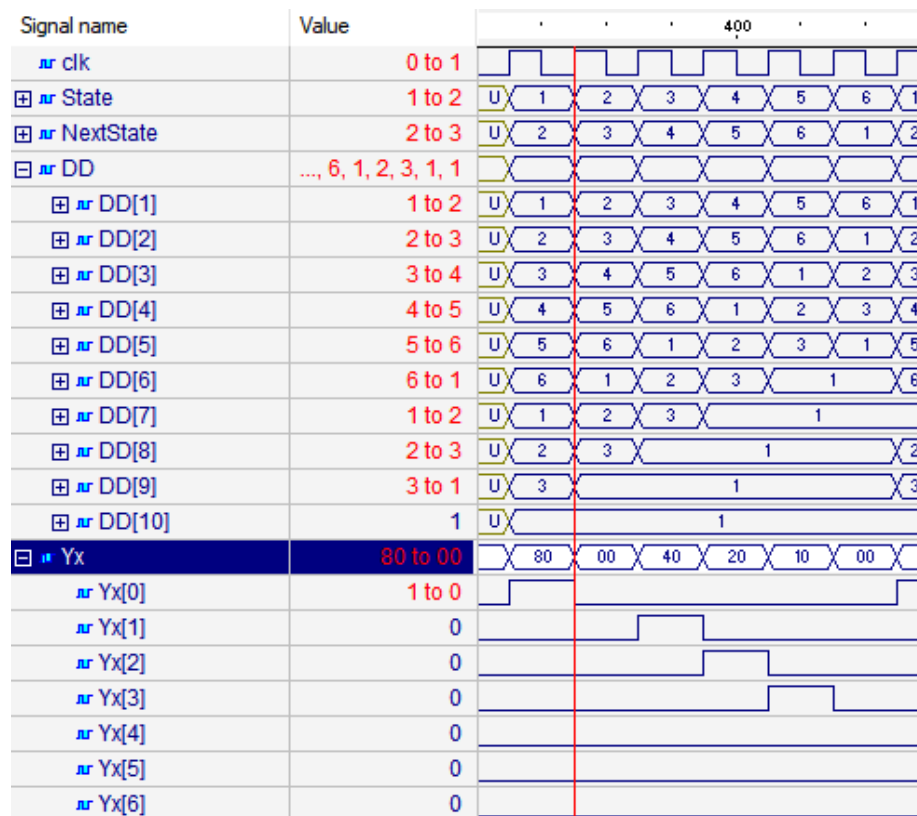



Рисунок Е,3 – Waveform ПД з використанням зсуву вхідної послідовності
(для А1-А2-А3-А4-А5-А6-А1)

ДОДАТОК Ж

РЕЗУЛЬТАТИ АВТОМАТИЗОВАНОГО СИНТЕЗУ VHDL-МОДЕЛЕЙ КА
МПА СКЛАДАННЯЖ.1 Протокол автоматизованого синтезу інструментальними засобами САПР
XILINX ISE VHDL-моделей КА МПА складання

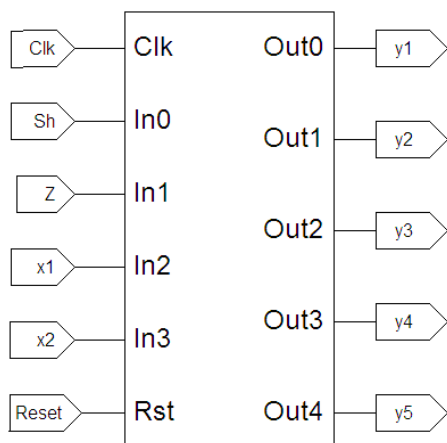
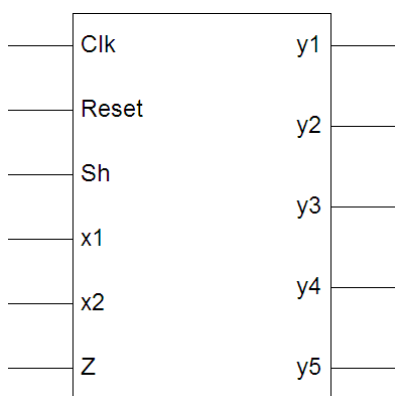
Analyzing FSM <FSM_0> for best encoding.

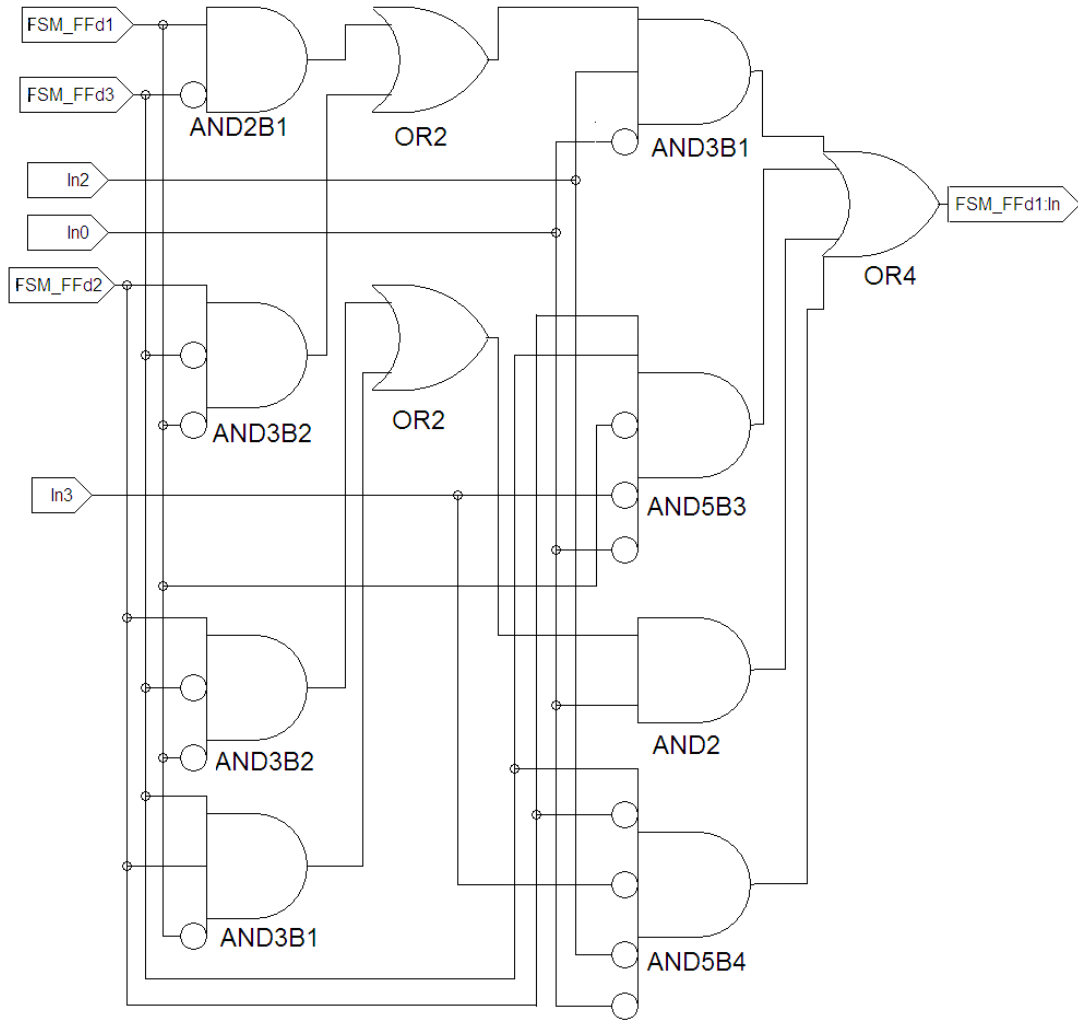
Optimizing FSM <State/FSM> on signal <State[1:3]> with gray encoding.

State | Encoding : a0| 000 a1| 001 a2| 011 a3| 110 a4| 010 a5| 111

В схемній реалізації використані наступні кодування: FSM_FFd1 – Q1, FSM_FFd2 – Q1, FSM_FFd3 – Q1, In0 – Sh, In2 – x1, In2 – x2, FSM_FFd1:In – Q1. В аналітичному вигляді функція збудження тригеру першого розряду має

ВИГЛЯД:





ДОДАТОК К

РЕЗУЛЬТАТИ АВТОМАТИЗОВАНОЇ ПОБУДОВИ
VHDL-МОДЕЛЕЙ КА КП ГРПЕ.1 Результати автоматизованої побудови VHDL-моделей КА КП ГРП
з використанням розробленого програмного модуля

```

library IEEE;
    use IEEE.std_logic_1164.all;
entity FSM is
    port (clk: in STD_LOGIC;
          X: in STD_LOGIC_VECTOR (0 to 9);
          reset: in STD_LOGIC;
          y: out BIT);
end FSM;
architecture arc of FSM is
type statetype is (a1, a2, a3, a4, a5, a6, a7, a8, a9);
signal state, nextstate: statetype;
begin
    State_NextState: process (X, State)
    begin
        case state is
            when a1 =>
                if (X(0)='0') then nextstate <= a1;
                else nextstate <= a2; y1 <= 1;
                end if;
            when a2 => nextstate <= a3; y2 <= 1;
            when a3 =>
                if (X(1)='0') then nextstate <= a1; y8 <= 1;
                else nextstate <= a4; y3 <= 1;
                end if;
            when a4 =>
                if (X(2)='0') then nextstate <= a1; y9 <= 1;
                else nextstate <= a5; y4 <= 1;
                end if;
            when a5 =>
                if (X(3)='0') then nextstate <= a5; y12 <= 1;
                else nextstate <= a6; y5 <= 1;
                end if;
            when a6 =>
                if (X(4)='0') then nextstate <= a1; y10 <= 1;
                else nextstate <= a7; y6 <= 1;
                end if;
        end case;
    end process;
end arc;

```

```

when a7 =>
    if (X(5)='1') then nextstate <= a8; y7<= 1;
    else nextstate <= a9; y <= 0;
    end if;
when a8 =>
    if (X(6)='1') then nextstate <= a1;
    else nextstate <= a1; y <= 0;
    end if;
when a9 =>
    if (X(7)='1') then nextstate <= a8; y7 <= 1;
    else nextstate <= a1; y14 <= 1;
    end if;
when others => NextState <= a1;
end case;
end process;
State_CurrentState: process (clk,reset)
begin
    if (reset='1') then state <= statetype'left;
    elsif rising_edge(clk) then state <= nextstate;
    end if;
end process;
end arc;

```

E.2 Результати автоматизованої побудови тестопридатної VHDL-моделі КА КП ГРП з використанням розробленого програмного модуля

```

library IEEE;
    use IEEE.std_logic_1164.all;
entity FSM is
    port (clk: in STD_LOGIC;
          X: in STD_LOGIC_VECTOR (0 to 9);
          reset: in STD_LOGIC;
          y: out BIT);
end FSM;
architecture arc of FSM is
type statetype is (a1, a2, a3, a4, a5, a6, a7, a8, a9);
signal state, nextstate: statetype;
begin
    State_NextState: process (X, State)
    begin
        case state is
            when a1 =>
                if (Sh='1') then nextstate <= a2;
                elsif (X(0)='0') then nextstate <= a1;
                else nextstate <= a2; y1 <= 1;

```

```

        end if;
    when a2 =>
        if (Sh='1') then nextstate <= a3;
        else nextstate <= a3; y2 <= 1;
        end if;
    when a3 =>
        if (Sh='1') then nextstate <= a4;
        elseif (X(1)='0') then nextstate <= a1; y8 <= 1;
        else nextstate <= a4; y3 <= 1;
        end if;
    when a4 =>
        if (Sh='1') then nextstate <= a5;
        elseif (X(2)='0') then nextstate <= a1; y9 <= 1;
        else nextstate <= a5; y4 <= 1;
        end if;
    when a5 =>
        if (Sh='1') then nextstate <= a6;
        elseif (X(3)='0') then nextstate <= a5; y12 <= 1;
        else nextstate <= a6; y5 <= 1;
        end if;
    when a6 =>
        if (Sh='1') then nextstate <= a7;
        elseif (X(4)='0') then nextstate <= a1; y10 <= 1;
        else nextstate <= a7; y6 <= 1;
        end if;
    when a7 =>
        if (Sh='1') then nextstate <= a9;
        elseif (X(5)='1') then nextstate <= a8; y7 <= 1;
        else nextstate <= a9; y <= 0;
        end if;
    when a8 =>
        if (Sh='1') then nextstate <= a1;
        elseif (X(6)='1') then nextstate <= a1;
        else nextstate <= a1; y <= 0;
        end if;
    when a9 =>
        if (Sh='1') then nextstate <= a8;
        elseif (X(7)='1') then nextstate <= a8; y7 <= 1;
        else nextstate <= a1; y14 <= 1;
        end if;
    when others => NextState <= a1;
end case;
end process;
State_CurrentState: process (clk,reset)
begin
    if (reset='1') then state <= statetype'left;
    elsif rising_edge(clk) then state <= nextstate;
    end if;
end process;
end arc;

```