

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи аналізу тестопридатності керуючих
автоматів за графовими моделями

(тема)

Виконав:

студент II курсу, групи СКСм-19-2
Трегуб Р.Р.
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

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

Освітня програма Спеціалізовані
комп'ютерні системи
(повна назва освітньої програми)

Керівник: доц. Кулак Е.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри АПОТ

(підпис)

Чумаченко С.В.

(прізвище, ініціали)

2021 р.

2

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)

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

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

Харківський національний університет радіоелектроніки

ЗАТВЕРДЖУЮ:

Зав.

кафедри

(підпис)

2

0

р

“ ” - .

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові

Трегубу Ростислав

(прізвище, ім'я, по

1. Тема роботи Методи аналізу тестопридатності керуючих автоматів за графовими
моделями

затверджена наказом“ 26 ” 03

2. Термін подання 25.05.20

3. Вихідні дані до

Мова опису апаратури VHDL

САПР XILINX ISE

ПЛІС Spartan 3E

4. Перелік питань, що

Аналіз предметної області і постановка задачі

Метод оцінки тестопридатності керуючих автоматів

Методи забезпечення тестопридатності керуючих автоматів

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 20 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	01.02.2021-15.02.2021	
2	Аналіз джерел з проблемної галузі	16.02.2021-25.02.2021	
3	Постановка задачі	26.02.2021-28.02.2021	
4	Розробка методів	01.03.2021-15.04.2021	
5	Проведення експериментів	16.04.2021-01.05.2021	
6	Оформлення матеріалів атестаційної роботи	02.05.2021-10.05.2021	
7	Подання атестаційної роботи керівникові та її попередній захист	11.05.2021-17.05.2021	
8	Подання атестаційної роботи на рецензування	18.05.2021-24.05.2021	
9	Захист проекту	25.05.2021-30.05.2021	

Дата видачі завдання 01.02.2021

Студент


(підпис)

Керівник роботи



(підпис)

доц. Кулак Е.М.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 67 с., 17 рис., 5 табл., 4 дод., 49 джерел.

КЕРУЮЧИЙ АВТОМАТ, ГРАФ ПЕРЕХОДІВ, ДІАГНОСТИЧНИЙ ЕКСПЕРИМЕНТ, ТЕСТОПРИДАТНІСТЬ, VHDL-МОДЕЛ

Метою дослідження є розробка методу аналізу тестопридатності керуючих автоматів при побудові функціонального тесту, і способу модифікації автомата для підвищення його тестопридатності.

Об'єктом дослідження є графова модель переходів (ГП) керуючих автоматів.

Предметом дослідження є методологія аналізу тестопридатності (розрахунку показників) керуючих автоматів і модифікація моделі автомата для збільшення його тестопридатності.

Методи дослідження – теорія графів, булева алгебра, прикладна теорія цифрових автоматів, методи технічної діагностики для генерації тестів та проведення діагностичного експерименту, засоби автоматизованого проектування цифрових автоматів для побудови, моделювання та синтезу автоматних HDL-моделей.

Наукова новизна полягає в подальшому розвитку методів оцінки і забезпечення тестопридатності КА та удосконалено алгоритм побудови дерева рішень для обходу станів ГП автомата при наявності циклів для організації ДЕ.

Практична значимість: розроблені методи призначаються для САПР тестопридатних пристроїв та систем, і дають можливість проектувати легко тестовані КА.

ABSTRACT

Master's thesis: 67 pages, 17 figures, 5 tables, 4 appendixes, 49 sources.

FINITE STATE MACHINE, STATE DIAGRAM, DIAGNOSTIC EXPERIMENT, TESTABILITY, VHDL-MODEL

The purpose of research: development of the FSM testability analysis method for the functional test pattern generation, and development of the approach of the FSM modifying for improvement its testability.

The object of research: graph models of finite state machines (FSM).

The subject of research: method for the analysis of testability (index calculation) of the FSM and modification of the model of the FSM for the testability improvement.

The methods of researching: the theory of graphs, Boolean algebra, applied theory of digital automata, methods of technical diagnostics for the generation of tests and carrying out a diagnostics experiment, automated design of FSM for the CAD system.

The scientific novelty consists of the general development of methods for assessing and improving the testability of the FSM and the algorithm for building the tree of solutions for bypassing the FSM states for diagnostic experiment organizing.

The practical significance: developed methods are used for CAD systems of test attachments and systems and can design and easily test the systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	10
ВСТУП.....	12
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ.....	13
1.1 Аналіз літератури з питання дослідження.....	13
1.2 Постановка задачі.....	30
2 МЕТОД ОЦІНКИ ТЕСТОПРИДАТНОСТІ КЕРУЮЧИХ АВТОМАТІВ.....	31
2.1 Алгоритм пошуку шляхів обходу ГП керуючого автомата при ДЕ	31
2.2 Розрахунок показників тестопридатності керуючого автомату... Для забезпечення діагностичного експерименту для путей обходу графу (см. підрозд. 2.1) можна використовувати показник складності шляху для вибору путі.....	38 41
3 МЕТОДИ ЗАБЕЗПЕЧЕННЯ ТЕСТОПРИДАТНОСТІ КЕРУЮЧИХ АВТОМАТІВ.....	47
3.1 Метод використання додаткового вхідного сигналу.....	47
3.2 Метод використання зсувного регістру.....	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	64
ДОДАТОК А.....	69
ДОДАТОК Б.....	78
ДОДАТОК В.....	89
ДОДАТОК Г.....	96

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

ВСТ– вбудоване самотестування

ГСА – граф-схема алгоритму

ГП – граф переходів

ДЕ – діагностичний експеримент

КА – керуючий автомат

КЧ – комбінаційна частина

ОА – операційний автомат

ПЛІС – програмовані логічні інтегральні схеми

САПР – система автоматизованого проектування

ТПР – тестопридатне проектування

ASIC – Application Specific Integration Circuit (замовні інтегральні схеми)

ATPG – Automatic Test Pattern Generation (автоматична генерація тестових наборів)

BILBO – Built-in Logic Block Observer (спостерігач вбудованого блоку)

BSDP – Block Structure of Design Project (блокова структура проекту)

CAD – Computer Aided Design (автоматизоване проектування)

CY – Controllability (керованість)

DFT – Design for Test (тестопридатне проектування)

FPGA – Field Programmable Gate Array (програмована вентильна матриця)

FSM – finite state machines (керуючий автомат з кінцевим числом станів)

HFG – Hierarchical Functional Graph (ієрархічний функціональний граф)

IEEE – Institute of Electrical and Electronics Engineers (інститут інженерів з електротехніки та електроніки)

JTAG – Joint Test Action Group (об'єднана робоча група по тестуванню)

LUT – Look Up Table (таблиця перетворень)

OY – Observability (спостережуваність)

RTL – Register transfer level (рівень реєстрових передач)

SoC – System-On-Chip (система на кристалі)

STIL – Standard Test Interface Language (стандартна мова тестового інтерфейсу)

TAM – Test Access Mechanism (механізм тестового доступу)

TADATPG – Testability Analysis for Deterministic ATPG (аналіз тестопридатості для системи автоматичної генерації детермінованих тестів)

TU – Testability (тестопридатність)

VHDL – VHSIC Hardware Description Language (мова опису апаратури для високошвидкісних інтегральних схем)

VHSIC – Very High Speed Integrated Circuit (високошвидкісні інтегральні схеми)

ВСТУП

Тестове діагностування цифрових схем на рівні кристала, друкованої плати або системи є дорогим процесом. Оцінки вартості тестування у відсотках від вартості повного життєвого циклу виробу змінюються від дуже низького значення (менше 10%) до надзвичайно високого (більше 60%) відповідно до етапу життєвого циклу і позицією того, кому необхідно знайти кошти для фінансування цього процесу. Традиційно етапи логічного проектування і генерації тестів діагностування розглядалися окремо – проектування передувало етапу синтезу тестів, і розробники цифрової апаратури не притягувалися до процесу генерації тестів. Рівень розвитку засобів автоматизованого проектування в даний час дозволяє досить просто спроектувати схему, яка в сучасному розумінні цього поняття фактично не є тестопридатною. Сьогодні визнається, що якщо в результаті проектування необхідно мати тестопридатний пристрій, то забезпечення тестопридатності має виконуватися в процесі проектування. Такий підхід дав початок розвитку численних методів тестопридатного проектування (ТПР).

Тестопридатність - це один з найбільш важливих показників, який повинен враховуватися при проектуванні цифрових пристроїв поряд з такими показниками, як швидкодія і вартість пристрою. Низький рівень тестопридатності виробу може підірвати шанси виробника на успіх при виході виробу на ринок. Існує безліч методів аналізу тестопридатності цифрових схем для логічного, функціонально-блокового і функціонального рівнів подання пристроїв, в тому числі орієнтованих на детерміноване, псевдовипадкове, зважене і функціональне тестування. Дана робота присвячена аналізу тестопридатності керуючих мікропрограмних автоматів по графові моделі для функціонального тестування автоматів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз літератури з питання дослідження

Загальні витрати, пов'язані з діагностичним забезпеченням виробу на етапі життєвого циклу, вважаються надмірно великими і, ймовірно, залишатимуться такими тривалий час. В процесі розвитку проектування цифрових пристроїв і систем було докладено значних зусиль, які були спрямовані на скорочення окремих видів витрат. Це зумовило розвиток методів проектування тестопридатості пристроїв і поява надійних і життєздатних методів проектування. Існує неформальне визначення тестопридатості [1]: схема є тестопридатною, якщо процедури генерації безлічі тестових наборів, оцінки їх ефективності та реалізації тестового діагностування можуть бути виконані за умови дотримання в встановлених межах фінансових витрат, витрат часу і значень показників, що характеризують пристосованість схеми до виявлення помилок, пошуку місця помилок і реалізації тестового діагностування.

Ключовим моментом цього визначення є необхідність виконання певних технічних вимог в межах допустимих фінансових витрат. Якщо реальна вартість тестового діагностування більше встановленої, то в відповідно до наведеного визначення виріб спочатку було спроектовано нетестопридатним. Зробити виріб тестопридатним можна або шляхом збільшення допустимих витрат, або шляхом скорочення вартості одного або декількох головних чинників, що визначають вартість програмного тестування. Якщо прийнятий другий підхід, то будь-яка процедура, яка забезпечує зменшення витрат на тестове діагностування, може розглядатися як процедура тестопридатного проектування.

Існуючі методи ТПР поділяються на три групи: спеціалізовані методи (ad hoc), методи сканування і методи вбудованого самотестування [1, 2, 3].

Методи сканування поділяються на структурні методи сканування і методи граничного сканування.

Неструктурні або спеціалізовані підходи (ad hoc) представляють собою сукупність правил поліпшення спостережуваності та керованості, що застосовуються на останніх етапах проектування. [1]. Додаткові апаратурні витрати, необхідні для реалізації цих методів відносно не великі (серед усіх методів ТПР - самі мінімальні апаратурні витрати).

Решта методів ТПР, на відміну від спеціалізованих, застосовуються, починаючи з самих ранніх етапів проектування, і додаткові апаратурні витрати, необхідні для реалізації цих методів набагато вище. Вони гарантують, що пов'язані з тестуванням проблеми виникатимуть, забезпечуючи покращений доступ до пристрою при тестуванні як для подачі тестових наборів, так і для зняття реакцій на тести. Для складних пристроїв, тільки змішане використання методів ТПР дасть найкращий результат забезпечення їх тестопридатості [1, 2].

Після спеціалізованих методів стали розвиватися структурні методи сканування. Суть їх полягає в тому, що існуючі елементи пам'яті, що входять в структуру пристрою, модифікуються і об'єднуються в єдиний зсувний регістр, іменованій шляхом сканування. Таким чином поліпшується керованість і спостережуваність внутрішніх вузлів схеми. При цьому схема виявляється розділеною на дві частини - послідовностну і комбінаційну. Кожна з цих частин тестується своїми способами. Для послідовностної частини використовуються стандартні тести для зсувного регістру, а для комбінаційної - детерміновані. Така ідея відвідала в середині 80-х років ряд компаній, які виробляють цифрову техніку. Кожна з компаній створила свій метод сканування. Додаткові витрати можуть доходити до 20%.

Крім великих витрат на реалізацію, методи сканування мають ще один істотний недолік - послідовний спосіб тестування, це вимагає щодо великого часу і обсягу пам'яті для зберігання тестів.

Природним розвитком методів структурного сканування з'явилися методи вбудованого самотестування (ВСТ) [1, 2, 3]. Суть принципу вбудованого самотестування в системах тестового (off-line) діагностування полягає в тому, що генерація тестових наборів і аналіз реакцій на ці набори здійснюється самим пристроєм. Для спрощення генератора тестової послідовності застосовують методи генерації, які характеризуються високою регулярністю, або, навпаки, випадкові тестові набори. Тестування проводиться на реальних частотах, що не менш важливо, оскільки непостійні несправності перевіряються або на робочій частоті пристрою, або на частоті, близькій до неї. Це одна з переваг, які дають методи ВСТ. Крім того методи не вимагають моделювання помилок, спрощують перевірку пристроїв, знижують час тестування, значно знижується потреба в зовнішньому діагностичному обладнанні.

У системах ВСТ використовуються вичерпні, псевдовичерпні і псевдовипадкові тести [3]. Вичерпні тести використовуються для комбінаційних схем, що містять не більше 22 входів, такі тести генеруються лічильниками. Псевдовипадкова послідовність може бути отримана за допомогою схеми, званої зсувними регістром з лінійними зворотними зв'язками (ЗРЛЗЗ). Проблема при використанні псевдовипадкових тестів полягає у визначенні довжини генерується послідовності тестових наборів для виявлення необхідного числа помилок. Для цього використовуються спеціальні методи аналізу тестопридатості. Всі ці способи тестування добре підходять тільки для комбінаційних схем. Для схем, що погано піддаються псевдовипадковому тестуванню будують детермінований тест. Детермінований тест в системах ВСТ зберігається в постійних запам'ятовуючих пристроях (ПЗП).

Методи стиснення вихідних реакцій характеризуються складністю реалізації та ймовірністю втрати інформації про несправності за рахунок стиснення. Для стиснення використовується дві структури - лічильник і ЗРЛЗЗ. Більш популярним при ВСТ є ЗРЛЗЗ, в такій реалізації його

називають сигнатурним аналізатором. Існує два види сигнатурних аналізаторів - послідовний або одновходовий і паралельний або багатовхідний [2].

Одним з найвідоміших методів ВСТ є метод BILBO (Built-in Logic Block Observer) [4]. Його зміст полягає в перетворення елементів пам'яті схеми в регістри BILBO для тестування комбінаційних схем, розташованих між ними. BILBO регістр являє собою багатофункціональний пристрій, що дозволяє генерувати псевдовипадковий тест, виконувати функції паралельного і зсувного регістру, здійснювати стиснення реакцій на тест послідовним або паралельним способом. Додаткові витрати можуть доходити до 30%.

При появі багат шарових друкованих плат в середині 80-х років гостро постала проблема їх тестування, оскільки до цього тестування друкованих плат ґрунтувалося на прямому доступі до провідників і мікросхем [5]. Для вирішення цієї проблеми була створена група JTAG (Joint Test Action Group), яка за допомогою 200 провідних фірм протягом 5 років розробила стандарт Boundary-Scan IEEE 1149.1 [6]. В основі методу лежить концепція розміщення послідовного зсувного регістру по межах пристрої. Кілька пристроїв на платі можуть бути об'єднані в єдиний послідовний шлях сканування. Таким чином, метод граничного сканування забезпечує практично 100% спостережуваність, і 100% керованість входів/виходів пристроїв. Основним завданням архітектури Boundary-Scan, відповідно до стандарту IEEE 1149.1 є тестування з'єднань (перевірка наявності, розміщення і зв'язку з'єднаних компонентів, а також виявлення обривів з'єднань і замикань між пристроями). Переваги технології Boundary-Scan: забезпечує простоту управління конфігурацією складних систем, що включають дочірні плати, багатокристалльні модулі і т.п. дозволяє організовувати on-line тестування; забезпечує доступ до регістрів, шинам, висновків; простий доступ до BIST можливостям пристроїв; полегшує тестування не пристосованих до сканування пристроїв, пам'яті. Перша

редакція стандарту IEEE 1149.1 була прийнята в 1990 році, нова версія - в 2013 році [7].

Завдяки високому рівню розвитку інтегральних технологій в останні роки досить популярними стали системи на кристалі SoC (System-On-Chip). Для організації їх тестування був розроблений стандарт IEEE 1500 Standard for Embedded Core Test (для тестування вбудованих ядер) [8]. За своєю структурою SoC нагадує структуру плати. Тому група розробників стандарту IEEE 1500 запропонувала адаптувати метод граничного сканування (IEEE 1149.1) для вирішення проблем тестування SoC. Однак, існує важлива відмінність. На рівні плати, можна робити припущення, що пристрій, встановлений на ній протестовано окремо до установки, і немає необхідності перевіряти його працездатність з використанням засобів граничного сканування. У стандарті IEEE 1149.1 команда INTEST є обов'язковою.

На SOC рівні це припущення не вірно. Навіть модуль «жорсткого» ядра є «м'яким», поки нарощування кристала не завершено. Це означає, що будь-яка додаткова тестова структура повинна передбачати функціональну перевірку модуля також як і структурну (перевірку виробничого дефекту). Команда INTEST (для тестування функціонування) обов'язкове, так як ядра необхідно протестувати після їх додавання на кристал. Тому були запропоновані багаторівневі структури граничного сканування дуже схожі на IEEE 1149.1 структури. Ці структури називаються wrappers (дослівно - обгортки) і підтримують внутрішньо- та зовнішньо-ядерні функції тестування, реалізовані за допомогою механізму тестового доступу (TAM - Test Access Mechanism). Сам TAM точно не визначений стандартом, але спосіб визначення тестів специфікований. Загалом, тести це комбінації векторів і протоколів (правил їх проходження). Рекомендується також використовувати для побудови тестів інший стандарт - IEEE 1450 Standard Test Interface Language (STIL) - (стандартна мова тестового інтерфейсу). Стандарт STIL був затверджений в 1999 році, нова версія - в 2006 [9]. Він присвячений проблемі проміжного формату для тестів, згенерованих за

допомогою засобів автоматичної генерації тестових наборів, наприклад, для тестів, які повинні транспортуватися через АТЕ (Automatic Test Equipment - автоматичне випробувальне обладнання) середу. Стандарт IEEE 1500 був прийнятий в 2006 році.

Розширенням SoC є мережі на кристалі Network-on-Chip (NoC). Це нова альтернатива, яка долає вузькі місця для інтеграції великої кількості ядер на одному SoC [10]. Зв'язок ядер з іншими компонентами SoC здійснюється через керовану шину даних. Шинна організація має ряд недоліків, таких як зниження швидкодії системи, проблеми, пов'язані з комунікацією вкладених модулів при великій кількості модулів та ін., Що дало поштовх до пошуку нових рішень комунікації компонентів. Стандарт IEEE 1687 [11], який є розширенням IEEE 1500 на нові комунікаційні архітектури рекомендує структуру оболонки аналогічну IEEE 1500. Як і IEEE 1500, IEEE 1687 визначає, як створювати узгоджену NoC, не нав'язуючи при цьому спеціальний інтерфейс для пристрою.

Методи аналізу тестопригодності розроблялися для різних рівнів опису цифрового виробу, в залежності від тенденцій розвитку засобів проектування.

Перші роботи в цій області належать Рутману [12] 1972 р. Незалежно від нього велися дослідження Стефенсоном і Грасоном [13] 1976 р., [14] 1979 р., [15] 1981 р. Ці роботи були орієнтовані на детерміноване тестування і присвячені системі TMEAS. В системі TMEAS значення спостережуваності та керованості нормалізуються в інтервалі від 0 (низький рівень) до 1 (високий рівень). Для обчислення значень керованості схеми СУ її входів приписують значення $SU = 1$. Потім залежність керованості інших вузлів схеми від станів їх входів представляється системою рівнянь. У цих рівняннях використовуються коефіцієнти передачі керованості елементів, які визначають співвідношення між значеннями керованості входів і виходів елемента схеми. Коефіцієнти передачі керованості змінюються в залежності від логічної функції компонента схеми і є функцією однорідності входів-

вихідного відображення. Коефіцієнт передачі керованості компонента схеми дорівнює 1, якщо вхід-вихідна відображення однорідно, і зменшується до 0 у напрямку зниження однорідності.

При обчисленні спостережуваності схеми ОУ спочатку передбачається, що для первинних виходів $O_U = 1$. Потім значення ОУ інших вузлів схеми зв'язуються зі значеннями ОУ виходів за допомогою коефіцієнтів передачі спостережуваності. Ці коефіцієнти визначають міру ймовірності того, що несправність, що з'являється на входах компонента схеми, буде поширюватися на її виходи, і знову значення коефіцієнтів змінюються в межах від 0 до 1 відповідно до виконуваної логічною функцією компонента. В системі ТМЕАС обчислення спостережуваності не залежить від значень керованості, обчисленої раніше. Для обчислення СУ і ОУ компонент схем, що містять елементи пам'яті, тактові шини і розгалуження, розроблені спеціальні правила. Коротко вони полягають у наступному. Елемент пам'яті представляється у вигляді елемента з додатковими лініями зв'язку, які визначають залежність подальшого стану елемента від його поточного стану. Наявність шин тактової синхронізації не враховується. При обчисленні керованості в вузлах розгалужень значення СУ зменшується в порівнянні зі значенням СУ при прямій передачі інформації, відображаючи той факт, що в точках розгалуження можна встановити різні логічні значення, як часто потрібно для активізації шляху транспортування несправності. Ці правила відрізняються від використовуваних в системі CAMELOT. Значення тестопридатості обчислюється як середнє арифметичне значень спостережуваності та керованості.

Обмеженість системи ТМЕАС полягає в тому, що задані формули для обчислення значень коефіцієнтів передачі спостережуваності та керованості за функціональними характеристиками елементів визначають тенденцію до отримання надзвичайно низьких значень цих параметрів для найпростіших схемних елементів (І, АБО та ін.), проте це не є серйозним недоліком, так як завжди при необхідності можна збільшити і значення коефіцієнтів до

необхідного рівня. Ще одне ускладнення, загальне для систем TMEAS і CAMELOT, виникає, коли зустрічаються вузли схеми, постійно підключені до одного з логічних рівнів. Виникає воно з тієї причини, що обчислення одного значення керованості для кожного вузла має тенденцію приховувати відносні труднощі формування двох можливих логічних рівнів. Рішення цієї проблеми запропоновано в наступних трьох системах оцінки спостережуваності.

Робота Рутмана була вдосконалена і розширена Брейером [16] 1978 р., [17] 1979 р. Ці роботи були орієнтовані на детерміноване тестування і присвячені системі TEST/80.

В системі TEST/80 пропонується програма автоматичної генерації тестів для цифрових схем. Подібно D-алгоритму в ній реалізується спроба встановити безліч шляхів, за якими несправності внутрішніх вузлів можуть транспортуватися через схему і спостерігатися на її виходах. Процедура виконується за дві фази: визначення шляху від вузла до виходу схеми і встановлення безлічі вхідних умов, що сприяють активізації цього шляху. На відміну від D-алгоритму в системі TEST/80 використовується процес, званий аналізом витрат, який дозволяє ідентифікувати оптимальний шлях переходу в будь-який стан. Насправді цей аналіз являє собою процедуру оцінки властивостей спостережуваності та керованості схеми.

Для кожного вузла, схеми обчислюються значення трьох оцінок: c_A - витрати на установку в вузлі А стану 1; c - витрати на установку в вузлі А стану 0; d_A -витрати на D-проход від вузла А до виходу схеми. Витрати зростають, коли виникають різні проблеми, що ускладнюють виконання цих процедур.

У загальному випадку описуються далі заходи систем SCOAP і TESTSCREEN дозволяють обчислити значення керованості вузла окремо для кожного логічного рівня (в даній системі c_A і c). Якщо відома функція елемента, то оцінки цих витрат на виходах деякого елемента обчислюються підсумовуванням наступних трьох складових: а) функції витрат керованості

входу; б) складовою, обумовленою несприятливими ефектами, які можуть виникати при наявності розгалужень на виході елемента; в) складової, пов'язаної з типом елемента. Завершивши обчислення оцінок цих складових для кожного елемента схеми, можна скласти систему рівнянь, з якої для всіх вузлів схеми визначаються значення sA і c .

Третя оцінка, dA , є мірою спостережливості вузла. Значення цієї оцінки на вході елемента знову обчислюється підсумовуванням трьох складових: а) оцінки ймовірності реалізації D-поширення зі входу елемента на його вихід, яка є функцією sA і c інших входів елемента; б) оцінки витрат на D-поширення від виходу елемента до виходу схеми; якщо є кілька шляхів, то вибирається шлях з найменшими витратами; в) оцінки числа тактових циклів, необхідних для передачі значення D від вузла на вихід схеми. Може здатися, що витрати dA рівні 0, якщо A - вихід схеми. Зауважимо, що принцип використання значень керованості для обчислення спостережливості такий же, як в системі CAMELOT.

Ця система оцінок дозволяє обчислити параметри тестопридатності, значення яких зменшуються, якщо умови, від яких залежать ці параметри, забезпечуються з меншими витратами. Таким чином, щоб максимізувати тестопридатність, витрати необхідно мінімізувати. З іншого боку, можливість обчислення значень керованості окремо для кожного логічного рівня збільшує точність моделювання процесу генерації тестів і дозволяє вирішити труднощі, пов'язані з наявністю в схемі вузлів, постійно підключених до одного з логічних рівнів.

Одним з найбільш відомих методів аналізу тестопридатності є метод Голдштейна SCOAP (Sandia Controllability / Observability Analysis Program) [18] 1979р., [19] 1980 р.

Система SCOAP є розвитком системи TEST/80. Кожен вузол схеми характеризується шістьма параметрами: комбінаційними керованість 0 і 1, послідовних керованість 0 і 1 і комбінаційної і послідовних спостережувані. Вони визначаються наступним чином. Комбінаційні керованості, CCO і $CC 1$,

визначаються як мінімальне число вузлів, які повинні бути встановлені в певний стан, щоб в заданому вузлі вийшло логічне значення 0 або 1. Послідовності керованості $SC0$ і $SC1$ визначаються аналогічно як число «послідовностних вузлів», які необхідно встановити в певний стан. Послідовностний вузол - це вузол схеми, стан якого фіксується на один тактовий період. Таким чином, вузол, який необхідно підтримувати в стані логічної 1 протягом шести тактів для транспортування необхідного логічного значення до заданого вузла, оцінюється як шість послідовних вузлів. Для входів схеми $CC0 = CC1 = 1$, в той час як $SC0 = SC1 = 0$ за визначенням.

Комбінаційна і послідовностна спостережуваність, $C0$ і $S0$, визначаються як мінімальне число вузлів або послідовних вузлів відповідно, які повинні бути встановлені в певний стан, щоб несправність транспортувалася від її джерела до виходу схеми. За визначенням, спостережливості виходів схеми рівні 0.

Як і в системі TEST/80, спостережуваність та керованість входів і виходів елементів схеми пов'язані логічною функцією, яка реалізується цим елементом. Крім того, їх значення обернено пропорційні обчислюваним параметрам вузлів схеми.

В [20, 21] (1982 г.) був описаний модифікований варіант алгоритму SCOAP, який задовольняє вимогам проектування пристроїв на вентильних матрицях. Ця система, яка називається COMET (Controllability and Observability MEasure for Test), включає додатковий параметр тестопридатності, званий передбачуваністю. Передбачуваність є похідною керованості і характеризує міру здатності чи нездатності схеми до початкової установці в відомий стан.

Інші варіанти базового алгоритму SCOAP забезпечують можливість більш досконалого поводження з шинами живлення і землі, вузлами з постійними логічними рівнями, двонаправленими елементами і макрокомірками.

Описані вище розробки, лягли в основу інших систем, які обчислювали значення спостережливості і керованості для детермінованих ATPG (Automatic Test Pattern Generation - автоматична генерація тестових наборів) систем, таких як TESTSCREEN [22] 1979 р., [23] 1981 р., CAMELOT (Computer-Aided Measure for Logic Testability) [24] 1981 р. і VICTOR (VLSI Identifier of Controllability, Testability, Observability and Redundancy) [25] 1982 р.

Система TESTSCREEN. Один з недоліків системи SCOAP полягає в тому, що для лінії з n інверторів керованість виходу дорівнює $n-1$. Однак реально управляти виходом n -го інвертора НЕ набагато важче, ніж виходом першого. Для виключення цього недоліку в системі TESTSCREEN змінено визначення спостережливості, значення якої визначається числом входів схеми, які повинні бути встановлені в певний стан для досягнення бажаного результату. Так, наприклад, комбінаційна керованість визначається як число входів схеми, які повинні бути встановлені в певний стан для отримання необхідного логічного значення на заданому вузлі. Отже, для лінійки інверторів комбінаційна керованість виходу дорівнює тепер 1. тестопридатності в системі TESTSCREEN визначається як зважена функція шести показників спостережуваності та керованості, розміру схеми і числа входів і виходів.

Система CAMELOT розроблена спільно Брюнельським університетом, (Аксбрідж, Великобританія) і компанією Cirrus Computers. В системі CAMELOT керованість СУ може приймати відносно значення від 0 до 1. Максимальне значення 1 має такий вузол, як первинний вхід, де можна легко встановити як логічну 1, так і логічний 0. Інша граничне значення керованості 0 має вузол, який не може бути встановлений в одне з двох можливих логічних станів. Практично значення керованості більшості вузлів у схемі лежать між цими двома межами. Визначивши значення СУ на первинних входах, визначаю значень СУ внутрішніх вузлів. Якщо входи пристрої управляються безпосередньо, то керованості його виходів повинні

просто в відносних одиницях відобразити міру здатності пристрою до установки на кожному виході 0 і 1, що визначається логічною функцією передачі пристрою. Однак в загальному випадку керованість входів пристрою не 100%. Тому керованість виходів повинна враховувати як здатність до передачі логічних значень через пристрій, так і значення керованості на його входах. Таким чином, вираз, що використовується для обчислення значення SU кожного виходу, використовує коефіцієнт передачі керованість, який є мірою, що характеризує ступінь відмінності здатності пристрою генерувати на даному виході значення 1 від здатності генерувати значення 0. Цей коефіцієнт залежить тільки від логічної функції, яка реалізується пристроєм, і не залежить від місця його розташування в схемі.

Другий показник тестопридатності – спостережуваність ОУ обчислюється таким чином. Спостережуваність вузла визначається як міра, що характеризує здатність схеми до передачі інформації про логічному стані даного вузла на один або кілька її виходів. Це означає, що спостережуваність первинного виходу дорівнює 1. Аналогічно, спостережуваність вузла в самому вузлі також дорівнює 1. Це значення зменшується у міру проходження сигналів уздовж активізованого шляху до первинних виходів. Для обчислення спостережуваності використовують коефіцієнт передачі спостережуваності. Коефіцієнт передачі спостережуваності є кількісною мірою, що характеризує зменшення значення спостережуваності вхідного вузла пристрою в міру просування до виходу вздовж активізованого шляху за умови, що інші вхідні вузли пристрою керовані. Цей коефіцієнт визначає здатність пристрою передавати зміни логічного стану одного входу на певний вихід. Однак в загальному випадку процес поширення інформації про несправності через пристрій залежить як від умови активізації певного входу, так і від умови установки фіксованих значень на деяких або всіх інших входах пристрою, що дозволяють активізувати шлях до певного виходу. Очевидно, що здатність пристрою задовольняти другому з цих умов є функцією керованості цих входів.

Міра тестопридатності вузла ТУ виходить в результаті множення значень його спостережуваності та керованості.

Обмеженість системи CAMELOT, яка властива і системі TMEAS, полягає в тому, що задані формули для обчислення значень коефіцієнтів передачі спостережуваності та керованості за функціональними характеристиками елементів визначають тенденцію до отримання надзвичайно низьких значень цих параметрів для найпростіших схемних елементів (І, АБО), проте це не є серйозним недоліком, так як завжди при необхідності можна збільшити значення коефіцієнтів до необхідного рівня. Ще одне ускладнення, загальне для системи CAMELOT, виникає, коли зустрічаються вузли схеми, постійно підключені до одного з логічних рівнів. Виникає воно з тієї причини, що обчислення одного значення керованості для кожного вузла має тенденцію приховувати відносні труднощі формування двох можливих логічних рівнів.

Програма аналізу тестопридатності VICTOR була розроблена з метою ідентифікації надлишкових вузлів схеми до початку процесу генерації тестів. Вузол схеми є логічно надлишковим, якщо значення на виходах схеми не залежать від стану вузла для всіх вхідних наборів або послідовності станів. На практиці це означає, що певні несправності на логічно надмірному вузлі не можуть бути виявлені. Ідентифікація потенційно надлишкових елементів схеми перед процедурою генерації тестів дозволяє виключити непотрібні витрати часу і, в кінцевому рахунку, даремні обчислення.

Система VICTOR аналізує комбінаційні підсхеми, в яких є сходяться розгалуження і реалізовані методи сканування. Програма повідомляє про наявність у схемі надлишкового вузла на основі аналізу конфліктів логічних значень в цьому вузлі. Користувач потім може вибрати одне з двох рішень: або модифікувати схему, або дати команду генератору тестів не синтезувати тест для помилок надлишкового вузла.

Деякий час розвивалися ентропійні методи аналізу тестопридатності. В [26] запропонований метод аналізу тестопригодності, заснован на теоретико-

інформаційних оцінках і тому мало пов'язаний з методами аналізу, розглянутими раніше. В даному методі керованість і спостережуваність визначаються як змінні, подібні ентропії. Керованість входів є високою, так як відомо, які логічні сигнали будуть включені. У міру просування сигналів через схему, кількість інформації про логічному стані внутрішніх вузлів схеми убуває, а отже, зменшується керованість. Аналогічно спостережуваність виходів схеми є високою і зменшується в міру просування до первинних входів. Ці визначення добре узгоджуються з аналогічними, використовуваними в системах CAMELOT і TMEAS. Цей метод представляв виключно теоретичний інтерес, важко застосуємо на практиці, особливо для послідовностних схем.

Пізніше з'явився метод визначення загального коефіцієнта якості тестопридатності Р. Спіллмена та ін. (факультет математики та обчислювальної техніки лютеранського університету, Такома, США) [27], 1983 р. Він заснований на двох базових поняттях: тестопридатності вузлів знаходяться в схемі і тестопридатності ізольованих вузлів, тобто вузлів, що знаходяться поза схемою. Знаходження тестопридатності вузлів схеми засноване на тестопридатності ізольованих вузлів і структури схеми. Використовується поняття досяжності вузлового елемента схеми, яке характеризується доступністю всіх вузлів, пов'язаних з цим елементом на шляху проходження сигналу від входу до виходу. Визначення загального коефіцієнта якості тестопридатності зводиться до вирішення трьох завдань: визначення всіх шляхів від входів до виходів; обчислення тестопридатності для кожного шляху; обчислення коефіцієнта якості тестопридатності пристрою, який дорівнює зваженому середньому тестопридатності шляхів. Метод застосуємо для невеликих схем на увазі необхідність визначення всіх шляхів від входів до виходів.

У всіх описаних методах існували дві проблеми. По-перше, не була досягнута кореляція між значеннями тестопридатності і числом перевіряються помилок (якістю тесту). По-друге, залишилося нез'ясованим

як модифікувати схему, щоб підвищити тестопридатність. Тривіальне рішення, що складається в додаванні додаткових контрольних точок на лінії з поганою керованістю, а також схем, що підвищують керованість на лінії з поганою керованістю не завжди ефективно. В [28] (1985 г.) був запропонований більш складний підхід, але його обчислювальна складність занадто висока, щоб бути використаним на практиці. В цілому, більшість існуючих методів містять дуже складні обчислення, застосовні тільки для невеликих схем (сотні вентилів) і важко піддаються аналізу. Аналізовані методи виявилися неприйнятними навіть для відносно невеликих схем уже на першому етапі - підрахунку основних показників.

Дослідницької група розробників системи Sigetest в частині аналізу тестопридатності цифрових пристроїв запропонували методи, позбавлені ряду наведених недоліків. Метод TADATPG (Testability Analysis for Deterministic ATPG - аналіз тестопридатності для системи автоматичної генерації детермінованих тестів) [29-32] призначений для аналізу тестопридатності комбінаційних і послідовних асинхронних схем. Він дозволяє розраховувати показники спостережуваності та керованості, до етапу побудови детермінованого тесту, з метою отримання його максимальної якості. Метод також пропонує спосіб вибору контрольних точок в схемі по розраховується показниками і спосіб модифікації схеми. До переваг методу можна віднести: відсутність втрати інформації при розрахунку показників тестопридатності для схем, тестованих детермінованим тестом; простота методу, при аналізі послідовних схем (не потрібно вирішувати систему лінійних рівнянь); метод забезпечує простий спосіб вибору ділянок схеми для модифікації.

До недоліків методу можна віднести: збільшення часу тестування за рахунок чергування режиму нормального функціонування і режиму сканування тестових даних в осередках вбудованого регістра для просування частини тестового вектору до внутрішніх точок схеми, і зсуву відгуків на

тест до спостережуваного виходу; зниження швидкодії схеми за рахунок введення додаткових апаратних засобів.

В [33] описується імовірнісний метод аналізу тестопридатності на рівні регістрових передач. Метод заснований на методі Паркера-МакКласкі [34] і використовується при аналізі довічних діаграм рішень (Binary Decision Diagrams). Результати по покриттю помилок істотно не відрізняються від результатів, отриманих на більш точному рівні уявлення - вентильному рівні опису. Розроблені методи і алгоритми засновані на розповсюдженні сигналів в схемі, представленій діаграмою рішень.

У роботах [35, 36, 37] представлені методи імовірнісного аналізу тестопридатності - розрахунку показників керованості, спостережливості і тестопридатності на вентильному рівні для псевдовипадкового тестування, а також введення керуючої логіки для важкодоступних вузлів. Ліній з найгіршими показниками спостережуваності та керованості вказують на лінії в схемі, які вимагають додаткової перевірки. Пропонується використання генератора зваженого тесту, побудованого таким чином, що обрані лінії схеми будуть безпосередньо керованими, тобто на лінію з низькою керованістю нуля буде подаватися тестова послідовність, що містить в собі переважно логічний нуль. У свою чергу, лінії, мають низьку керованість одиниці будуть безпосередньо керовані логічною одиницею.

Метод в роботі [38] заснований на імовірнісному підході обчислення показників тестопридатності вузлів пристрою, що дозволяє обчислювати керованість і спостережуваність виходів комбінаційних пристроїв рівня регістрових передач (шифраторів, суматорів, мультиплексорів та ін.). Також пропонується використання генератора зваженого тесту.

У статті [39] представлена стратегія процедури проектування SoC, що тестується. Цей підхід заснований на аналізі тестопридатності на різних рівнях абстракції: на вентильному рівні, на рівні регістрових передач, на системному рівні (для функціонального тестування FSM). Основна мета

представлених алгоритмів - збільшити покриття несправностей перед генерацією тестів і скоротити час перевірки.

В роботі [40] представлені функціональні методи тестопридатного проектування цифрових автоматів. Для автоматів, які задаються у вигляді таблиць переходів вводяться поняття класів автоматів і пропонуються методи приведення таблиць переходів автоматів до заданих класів. Розглянуто розпізнають, діагностичні, синхронізуючі і встановлюючі послідовності для їх використання в діагностичному експерименті. При функціональному тестуванні цифрових автоматів підвищення їх тестопридатності можливо тільки при введенні апаратною надлишковості шляхом розширення або вихідного алфавіту, або вхідного алфавіту, або алфавіту станів. З одного боку це призводить до збільшення витрат апаратури, але з іншого боку зберігає алгоритм функціонування автомата, заданого таблицею переходів або графом переходів.

В роботі [41] розглянуто питання проведення діагностичного експерименту над автоматами за допомогою обходу всіх дуг і вершин графа переходів. Для різних варіантів обходу графа було дано оцінки повноти і довжини діагностичного експерименту. Запропоновані процедури для підвищення тестопридатності цифрових автоматів. Розглянуто процедури розширення вхідного алфавіту і еквівалентних перетворень.

У системах управління під досяжністю розуміють міру можливості переведення системи в необхідний технічний стан [42]. Для аналізу складності граф-схем алгоритмів в [43] введено поняття досяжності. Показники досяжності використовуються для визначення місць в опису автомата з метою встановлення туди конструкцій з асерціями при виконанні верифікації [44]. В [45] розглянуто методи розрахунку спостережливості і керованості при аналізі графової моделі транспортного потоку для оптимізації маршрутів дорожнього руху.

1.2 Постановка задачі

Метою дослідження є розробка методу аналізу тестопридатності керуючих автоматів при побудові функціонального тесту, і способу модифікації автомата для підвищення його тестопридатності.

Об'єктом дослідження є граф переходів (ГП) керуючих автоматів.

Предметом дослідження є методологія аналізу тестопридатності (розрахунку показників) керуючих автоматів і модифікація моделі автомата для збільшення його тестопридатності.

Методи дослідження – теорія графів, булева алгебра, прикладна теорія цифрових автоматів, методи технічної діагностики для генерації тестів та проведення діагностичного експерименту, засоби автоматизованого проектування цифрових автоматів для побудови, моделювання та синтезу автоматних HDL-моделей.

Для досягнення поставленої мети слід вирішити такі задачі:

- розробити метод аналізу тестопридатності цифрових автоматів, представлених у вигляді орієнтованого графа для спрощення завдання генерації тестів;
- удосконалити алгоритм побудови дерева рішень для обходу станів ГП автомата при наявності циклів для організації ДЕ;
- розробити методи підвищення тестопридатності керуючих автоматів;
- реалізувати VHDL - моделі для проведення порівняльного аналізу розроблених методів.

2 МЕТОД ОЦІНКИ ТЕСТОПРИДАТНОСТІ КЕРУЮЧИХ АВТОМАТІВ

2.1 Алгоритм пошуку шляхів обходу ГП керуючого автомата при ДЕ

У сучасних системах автоматизованого проектування цифрових пристроїв проєктований пристрій зазвичай задається у вигляді алгоритму функціонування, описаного мовою опису апаратури. Для такої форми подання пристроїв застосування структурних методів діагностування, що використовують модель константних несправностей, стає неефективним з одного боку. З іншого боку, використання функціональних методів тестування часто є дуже складним через розмір сучасних пристроїв що проєктуються.

Одним з найбільш поширених механізмів управління та обробки даних є кінцевий автомат. Формами його подання є таблиця переходів-виходів і граф переходів. Через високу складність діагностичних експериментів (ДЕ) з автоматами, були запропоновані методи забезпечення тестопригодності, які передбачають введення апаратурної надмірності. На структурному рівні це введення додаткових входів і виходів для спрощення проведення ДЕ. На функціональному рівні це внесення змін до функціонального опису автомата, тобто в його таблицю переходів. Це дозволяє зробити тестопридатний автомат легкотестованим, тобто таким автоматом, для якого завдання тестового діагностування вирішуються максимально просто в рамках передбачуваних витрат на проєктування.

Способом опису моделей цифрових пристроїв у вигляді кінцевих автоматів на мові VHDL є автоматний шаблон. Це особлива структура VHDL-моделі, в якій функції виходів і переходів винесені в окремі процеси (або процес), а призначення нового стану реалізується в спеціально виділеному процесі під сигналом синхронізації. Апаратурну надмірність, яка забезпечує легкотестованість, доцільно вносити на початковому етапі

проектування, тобто до побудови мовних моделей проєктованих пристроїв. Легкотестованим будемо називати пристрій (кінцевий автомат), для якого можна побудувати діагностичний експеримент мінімальної довжини шляхом забезпечення установки пристрою (автомата) за мінімальний час в будь-який стан.

Схема є тестопридатною, якщо генерація множин тестових наборів, оцінка їх ефективності та реалізація тестового діагностування можуть бути виконані за умови дотримання в встановлених рамках фінансових і тимчасових витрат, а також значень показників, що характеризують здатність схеми до виявлення несправностей, локалізації несправностей і реалізації тестового діагностування [1].

Під експериментом над автоматом будемо розуміти процес подачі вхідних впливів, спостереження відгуків на ці вихідні впливи і формування висновку, заснованого на цих спостереженнях. Залежно від того, яка мета експерименту розрізняють експерименти по ідентифікації вхідної послідовності автомата і ідентифікації автомата з n станами, що відрізняється від всіх інших автоматів з таким же числом станів. Для виявлення помилок проєктування в моделях керуючих автоматів, представлених таблицею переходів і виходів або графом переходів, використовується діагностичний експеримент, що складається в обході всіх станів (вершин графа). У зв'язку з цим легкотестованим вважатимемо автомат, який може бути встановлений в будь-який стан за n або менш тактів, де n - число станів автомата (вершин графа) без використання синхронізуючих послідовностей [47].

Якщо розглядати HDL-моделі кінцевих автоматів у вигляді автоматного шаблону як об'єкт діагностування, то щоб побудувати легкотестовані автомати на їх основі, ці моделі повинні відповідати таким умовам:

- в моделі повинна бути реалізована можливість установки автомата в режим тестування і назад в режим нормального функціонування в будь-якому такті роботи автомата;

- в режимі тестування повинна бути реалізована можливість встановлення автомата в будь-який стан за $(n-1)$ тактів, де n - число станів автомата, і може бути забезпечений гамільтонов цикл проходження по автомату від будь-якого зі станів автомата;

- легкотестованій автомат повинен будуватися засобами САПР в автоматизованому режимі.

Далі будемо розглядати синхронні автомати, робота яких тактується сигналами від генератора тактових імпульсів (CLK), тобто крім входів (x_1, \dots, x_n) в автоматі є ще один вхід CLK, по якому проходить синхронізуюча послідовність. У зв'язку з цим вхідним сигналом на переході $a(t) \rightarrow a(t+1)$, відповідним шляху $a(t) X(a(t), a(t+1)) - Y(a(t), a(t+1)) a(t+1)$, буде не $X(a(t), a(t+1))$, а кон'юнкція $CLK \cdot X(a(t), a(t+1))$. Таким чином, при безумовному переході $X(a(t), a(t+1)) = 1$ автомат здійснює перехід тільки від синхросигналу. Але при формуванні графа переходів автомата сигнал CLK на дугах графа не зображується.

Для опису графа переходів автомату запропоновано використовувати модифіковану матрицю суміжності розміром $(M \times M)$, де M – кількість вершин в графі автомата Мура. Дана матриця характеризується тим, що для кожної вершини графа переходів (строки матриці) в комірках матриці знаходяться імена дуг, що з'єднують дану вершину з вершинами-наступниками (стовпцями). При цьому в одній комірці можуть бути більш однієї дуги (мультиграф). Слід відмітити, що умови переходу, які представлені декількома термами ДНФ, теж породжують декілька дуг і графі переходів. При цьому граф переходів має відповідати стандартним умовам досяжності: кожна вершина має бути досяжною з початкової вершини та з неї має бути путь до кінцевої вершини.

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 збудження 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 автомата, 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 аналізу результатів порівняння робиться висновок про відповідність HDL-моделі специфікації. ДЕ проводиться з використанням системи верифікації HDL-моделей (TestBench) в середовищах проектування Active-HDL або XILINX ISE.

Для реалізації стратегії обходу всіх дуг графа за методикою, запропонованою в [46], на підставі модифікованої матриці суміжності будується дерево рішень для обходу шляхів (маршрутів) графа (рис. 2 а, б у [46]).

У лінійній ГСА, де логічні умови слідуєть одна за одною, максимальна кількість шляхів обходу графа автомата Мура буде, 2^n , де n – кількість логічних умов x_i з двома альтернативами. При цьому умова одноразового покриття всіх дуг графа при його обході на даному етапі не аналізується.

Особливістю даного дерева є те, що термінальною вершиною у всіх маршрутах обходу графа КА є стан a_0 , тобто кожен шлях в графі реалізує повний цикл роботи ОА.

Для реалізації стратегії обходу всіх дуг графа (в режимі «ручної» побудови тестів) будується бінарне дерево рішень (дерево обходу графа) за наступним алгоритмом.

а) Побудова дерева обходу графа починається з початкової вершини a_0 . Аналізується перший рядок матриці суміжності і складається список вершин-наступників.

б) Зі списку наступників обирається перша вершина і включається в

дерево. Відповідна комірка в матриці (дуга) позначається.

в) Для рядка, що відповідає цьому номеру, складається список x наступників. При подальшому аналізі списку наступників перевага віддається непоміченим коміркам-наступникам.

г) Побудова маршруту в графі завершується у наступних випадках :

– чергова вершина-наступник є початковою (термінальною) вершиною a_0 (завершений цикл роботи автомата);

– вихідна дуга є петлею (елемент головної діагоналі матриці суміжності);

– у маршруті з'являється вершина, яка вже присутня у цьому маршруті (цикл у графі переходів, фіксується початкова вершина циклу);

д) П. п. б), в) та г) повторюються до тих пір, поки всі маршрути не дійдуть до термінальної (кінцевої) вершини.

е) Петлі та маршрути з циклами заносяться до окремого стеку. Петлі можуть вставлятися у будь-які маршрути, де є відповідні вершини.

ж) Для маршрутів зі стеку циклів добудовуються маршрути повернення в початкову вершину шляхом використання фрагментів маршрутів з цієї вершини до термінальної вершини (такий маршрут завжди буде виходячи зі стандартних умов досяжності графа).

Для реалізації стратегії обходу всіх дуг ГП за методикою, запропонованою в [46], будується дерево рішень для обходу шляхів (маршрутів) графа, дана методика не дозволяє обробляти нелінійні ГСА, які породжують цикли в ГП.

Запропонований в даній роботі метод дозволяє обробляти нелінійні ГСА, які породжують цикли в ГП.

Розглянемо приклад автомата, представленого на рис. 2.1 у вигляді ГСА. Представлена ГСА має три умови переходів, п'ять входжень цих умов в ГСА, вісім станів і кілька зворотних гілок переходів, що породжують цикли. Розмітка ГСА відповідає мікропрограмному автомату Мура.

Рисунок 2.1 – ГСА автомата Мура

На рис. 2.2 представлений граф переходів, який побудований за ГСА автомата Мура, зображеного на рис. 2.1. ГП має 8 вершин по числу станів автомата. На ньому опущені керуючі сигнали, але вказані умови переходів і кожній дузі, що позначає перехід, ставиться у відповідність її ім'я у вигляді букви латинського алфавіту.

Рисунок 2.2 – Граф переходів автомата Мура

За ГП, представленому на рис. 2.2, будується матриця суміжності (табл. 2.1).

Таблиця 2.1 – Матриця суміжності автомата Мура

	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_0		a						
a_1		c	b					
a_2				d	f			
a_3			e			h		k
a_4			g			i		p
a_5	l						n	
a_6			t			m		r
a_7	s							

По матриці суміжності будується дерево рішень. На рис. 2.3 представлений деревоподібний ГП, на якому пунктиром позначені можливі цикли в графі, а блакитним кольором – варіанти виходу з циклу.

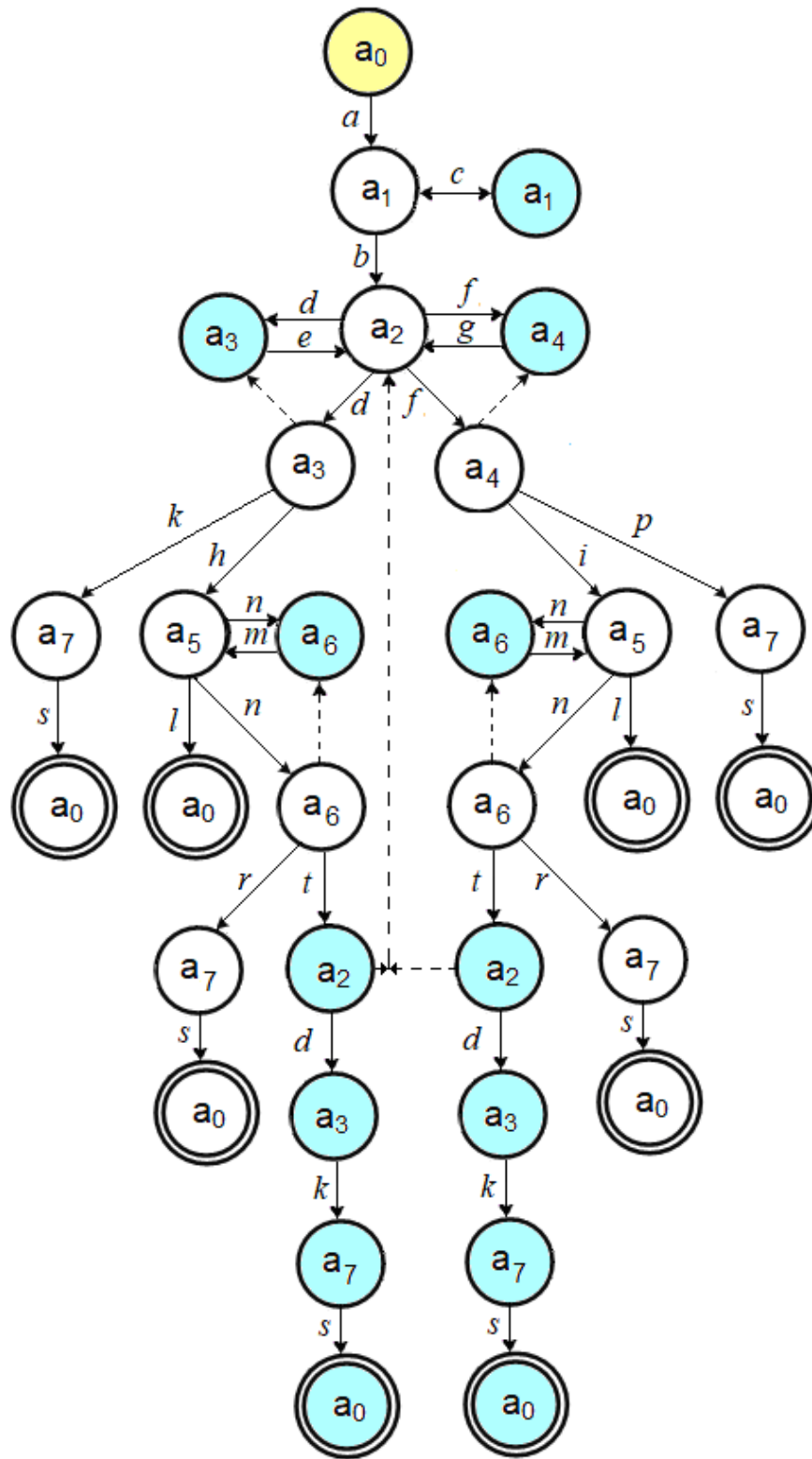


Рисунок 2.3 – Деревоподібний ГП

На рис. 2.4 представлено дерево рішень. Максимальне число шляхів у дереві буде 2^n , де n - не числом умов, а число входжень умов в ГСА.

Рисунок 2.4 – Дерево рішень

Для ГСА на рис. 2.1 $n = 5$ (5 входжень умов при трьох умовах) і відповідно верхня межа числа маршрутів буде дорівнює 32.

2.2 Розрахунок показників тестопридатності керуючого автомату

Під керованістю керуючих автоматів будемо розуміти здатність автоматів перейти в заданий стан. Керованість початкового стану дорівнює 100%, тобто початковий стан є безпосередньо керованим. Чим більше розглядається стан віддалене від початкового, тим нижче його керованість. На структурному рівні керованість буде залежати від функції збудження елементів пам'яті, що зберігають стану у вигляді кодів цих станів (функції переходів). Поліпшення керованості можна забезпечувати шляхом розширення вхідного алфавіту (збільшення числа вхідних сигналів).

Під спостережуваністю керуючих автоматів будемо розуміти здатність автоматів спостерігати поточний стан автоматів за вихідними (керуючими) сигналами. Спостережуваність залежить як від керованості, що забезпечує перехід в спостережуване стан так і від розрізнення вихідних сигналів сусідніх станів. Поліпшення спостережуваності можна забезпечувати шляхом розширення вихідного алфавіту (збільшення числа вихідних сигналів). На структурному рівні спостережуваність буде залежати від функції виходів і функції переходів. Тестопридатність автомата є функція спостережуваності та керованості. Поліпшення спостережуваності та керованості приводить до збільшення тестопридатності.

Оскільки спостережуваність і тестопридатність в цілому залежать головним чином від керованості, перш за все потрібно забезпечувати поліпшення керованості автомата.

Якщо до автомата не застосовується ніяких заходів по збільшенню його тестопридатності у вигляді модифікації його моделі, для проведення ДЕ необхідно використовувати методику, розглянуту в підрозділі 2.1.

Якщо застосовувати заходи по збільшенню тестопридатності необхідно визначитися зі способом оцінки керованості автомата.

Будемо розглядати керованість як досяжність. У системах управління під досяжністю розуміють здатність (складність, яка залежить від часу і дій, що ініціюють) системи встановлюватися в зазначений стан.

Під керованістю будемо розуміти ваговий коефіцієнт, який ставиться у відповідність кожній вершині графа переходів КА і в умовних одиницях характеризує складність установки автомата в потрібний стан, починаючи з початкового.

При цьому необхідно враховувати, що складність функцій виходів безпосередньо впливає на спостережуваність стану автомата, а для вирішення завдання перекладу автомата в необхідний стан складність функцій виходів можна не враховувати, так як розширення вхідного алфавіту (збільшення числа вхідних сигналів) впливає на складність реалізації функцій переходів, а функція виходів в даному випадку залишається незмінною.

Розрахунок показників керованості автомата будемо проводити шляхом аналізу графа переходів, в якому виконано кодування станів автомата.

Нехай в графі переходів автомата існує вершина a_n , в яку входять k дуг, тобто яка має k попередників (рис. 2.5) і відповідно до кожної k -й дузі графа походів будемо розраховувати її коефіцієнт керованості. Розрахунок коефіцієнтів буде враховувати на відміну від [43] не тільки сумарна вага умов переходу, а й відмінність двійкових кодів вершин автомата i . Розрахунок здійснюється за формулою:

$$, \quad (2.1)$$

де q_{m+k} – кодова відстань між двійковими кодами станів автомата a_{m+k} та a_n (якщо для зберігання станів використовуються тригери типу Т, JK), або кількість одиниць у вершини, куди входить дуга (якщо для зберігання станів використовуються тригери типу D), R – кількість умов переходу на дузі k , p_r – вага відповідної умови переходу.

Рисунок 2.5 – Коефіцієнти керованості КА

Варто зазначити, що в обчислювальних системах для мікропрограмних автоматів $p_r=1$. Для автоматів логічного управління p_r залежить від складності пристрою перетворення в логічні значення нуля та одиниці аналогових сигналів датчиків в залежності від діапазону заданих значень вхідних сигналів.

Керованість D_n вершини графа a_n обчислюється як мінімальна сума керованості вершин-попередників і коефіцієнтів керованості W_{m+k} для дузі $a_{m+k} \rightarrow a_n$:

(2.2)

Для початкової вершини a_0 керованість, слід відзначити, що петлі при розрахунку керованості не враховуються.

Метод обчислення керованості станів ГП.

а) Початкова вершина a_0 ГП помічається і їй присвоюється керованість $D_0=1$.

б) Для кожної вершини, що має метку обчислюються коефіцієнти керованості для всіх вихідних дуг W_j , де j – кількість вихідних дуг, за формулою (2.1), ці дуги також помічаються.

в) Для всіх вершин, які мають вхідні помічені дуги обчислюється керованість D_n за формулою (2.2). Ці вершини помічаються.

г) Пункти б) та в) слід повторювати доти, поки усі вершини графа не отримують мітки.

Для забезпечення діагностичного експерименту для путей обходу графа (см. підрозд. 2.1) можна використовувати показник складності шляху для вибору путі.

Як результат обчислення тестопридатності, показник складності p шляху S_p обчислюється за формулою:

$$S_p = \frac{H}{p}, \quad (2.3)$$

де H – кількість дуг для шляху p .

Для діагностичного експерименту з обходу графа переходів обирається той шлях, у якого S_p приймає мінімальне значення.

Для організації діагностичного експерименту (ДЕ) над графом керуючого автомату в загальному вигляді треба здійснити одноразовий обхід всіх дуг графа, що забезпечить перевірку всіх умов переходів в графі. Але граф, наведений на рис. 2.2, має дві особливості. По-перше це автомат Мура, тобто вихідні сигнали u_i віднесені тільки до вершин графа, а, по-друге, цей автомат належить до виключного класу автоматів, тобто всі його стани (вершини графа) мають різні комбінації вихідних сигналів. Таким чином, для проведення неруйнівного ДЕ достатньо здійснити одноразовий обхід всіх вершин графа, що відповідає організації одного або декількох гамільтонових циклів у цьому графі.

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 n – число вершин у графі ; якщо ступінь кожної вершини не

менше, ніж $\frac{p}{2}$, тобто $\frac{p}{2} - 1$, де p – найближче ціле число, то граф називається графом Дірака, який в свою чергу є гамільтоновим.

Перевіримо виконання умови Дірака. Для графа на рис. 2.2 ступені вершин наведені на рис. 2.6. Для цього графа умова Дірака виконується: $p = 7$, $\frac{p}{2} = 3.5$, для всіх вершин, окрім початкової, тобто $\deg(a_1) = 4$, $\deg(a_2) = 6 > 3.5$, $\deg(a_3) = 4$, $\deg(a_4) = 4$, $\deg(a_5) = 5 > 3.5$, $\deg(a_6) = 4$, $\deg(a_7) = 4$.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7
<i>deg</i>	4	6	4	4	5	4	4

Рисунок 2.6 – Ступені вершин

Виконання умови Дірака для початкової вершини не є обов'язковим тому, що за визначенням гамільтонів цикл неруйнівного ДЕ починається та закінчується у вершині a_0 (рис.2.7). Таким чином, граф на рис.2.2 є графом Дірака, тобто 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 – гамільтонов (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 – гамільтонов 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 .

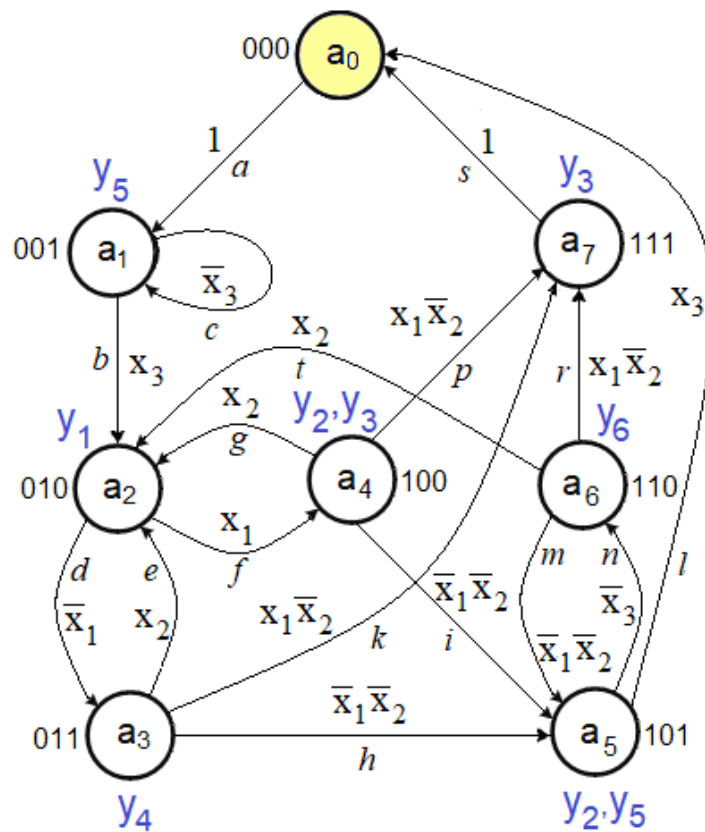


Рисунок 2.7 – Граф переходів автомата Мура

Таким чином, на основі теореми Бонді-Хватала для графа автомата для побудови множини гамільтонових циклів можна додати 4 пари несуміжних вершин, $(a_1 \leftrightarrow a_4)$, $(a_3 \leftrightarrow a_4)$, $(a_3 \leftrightarrow a_6)$, $(a_4 \leftrightarrow a_6)$.

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 циклу для якого функція переваги (вага p -го шляху) буде . При цьому слід враховувати, що вага додаткових дуг в графі дорівнює 1.

Проаналізувавши запропоновані додаткові дуги в графі можна зробити висновок, що тільки додатковий перехід $(a_3 \leftrightarrow a_4)$ дозволяє організувати гамільтонів цикл мінімальної ваги в цьому графі. Перевіримо це евристичне припущення математичним методом.

Серед способів визначення гамільтонових циклів у графі розглядаються методи перебору, алгебраїчний метод, а також методи гілок і границь, 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.

З урахуванням кодування по номеру вершини для графа на рис. 2.7 обчислимо коефіцієнти досяжності для всіх дуг та сформуємо матрицю суміжності розміру $[8 \times 8]$, яка задає вартість всіх попарних зв'язків між вершинами, причому відповідні елементи приймаються $c_{ij} = \emptyset$, якщо відсутні дуги між вершинами i, j . Матриця суміжності з ваговими коефіцієнтами дуг, розрахованими за формулою (2.1) для цього графу наведена в табл. 2.2.

Таблиця 2.2 – Матриця суміжності з ваговими коефіцієнтами дуг

	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_0		1						
a_1			2					
a_2				1	2			
a_3			1			4		2
a_4			2			2		4
a_5	2						2	
a_6			1			4		2
a_7	3							

Наведемо покрокове виконання методу гілок та границь для цього графу.

1. З кожного рядка матриці вартості віднімаємо мінімальний елемент (табл. 2.3):

Таблиця 2.3 – Матриця вартості

	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	min
a_0		0							1
a_1			0						2
a_2				0	1				1

a_3			0			3		1	1
a_4			0			0		2	2
a_5	0						0		2
a_6			0			3		1	1
a_7	0								3

2. Віднімаємо мінімальний елемент з тих стовпчиків, де немає нулів, тобто a_4, a_7 (табл. 2.4):

Таблиця 2.4 – Матриця вартості за вирахуванням мінімального елемента

	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_0		0						
a_1			0					
a_2				0	0			
a_3			0			3		0
a_4			0			0		1
a_5	0						0	
a_6			0			3		0
a_7	0							

Звертаємося до матриці, де мінімальні елементи з кожного рядка, а потім з кожного стовпця вже вирахувані. Ненульові числові елементи відповідають переходам $a_3 \rightarrow a_5$, $a_4 \rightarrow a_7$, $a_6 \rightarrow a_5$, яких слід уникати, бо вони роблять більший внесок у сумарний маршрут.

За матрицею складемо цикл $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_7 \rightarrow a_0$, який не є гамільтоновим, бо відсутні вершини a_4, a_5, a_6 . З матриці видно, що ці вершини між собою пов'язані переходами $a_4 \rightarrow a_5 \rightarrow a_6$. Оскільки з вершини a_6 є перехід до вершини a_7 , для існування гамільтонова циклу достатньо штучно додати дугу, яка б поєднувала обидва фрагменти циклу, тобто з вершини a_3 до вершини a_4 . Тоді маємо цикл $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow a_5 \rightarrow a_6 \rightarrow a_7 \rightarrow a_0$, що є гамільтоновим.

Другим способом за матрицею можна також скласти цикл

$a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_4 \rightarrow a_5 \rightarrow a_6 \rightarrow a_7 \rightarrow a_0$, який також не є гамільтоновим, оскільки відсутня вершина a_3 . В графі існує перехід з a_2 до a_3 , тому досить штучно додати дугу, що пов'язує вершини a_3 з a_4 , щоб поєднати між собою обидва фрагменти ланцюгів з метою утворення гамільтонова циклу. Тоді маємо цикл, $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow a_5 \rightarrow a_6 \rightarrow a_7 \rightarrow a_0$, який є гамільтоновим (рис.2.8).

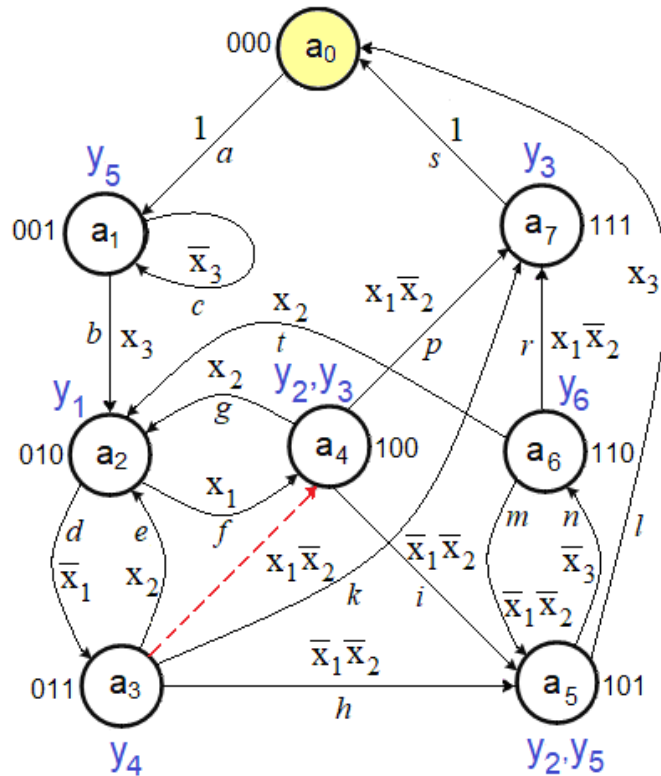


Рисунок 2.8 – Граф переходів автомата Мура з додатковою дугою $a_3 \rightarrow a_4$

Таким чином, отримали однакові гамільтонові цикли, що підтвердило наше припущення на основі евристичного методу.

3 МЕТОДИ ЗАБЕЗПЕЧЕННЯ ТЕСТОПРИДАТНОСТІ КЕРУЮЧИХ АВТОМАТІВ

3.1 Метод використання додаткового вхідного сигналу

Перший метод підвищення тестопридатності КА полягає в запровадженні додаткового вхідного сигналу в модель автомата, це відповідає додаванню додаткового стовпця в таблицю переходів або додаткових дуг в граф переходів автомата. Це дає можливість встановити автомат в потрібний стан не більше ніж за $(n-1)$ тактів, де n - число станів автомата. При автоматизованому синтезі таких автоматів було виявлено, що це призводило до зниження швидкодії і появи додаткових витрат апаратури, які не перевищують 30% при довільній розстановці додаткових переходів, що ініціюються додатковим вхідним сигналом.

Зниження додаткових витрат апаратури і підвищення швидкодії можна здійснити за рахунок організації певного способу обходу станів в автоматі при розстановки додаткових переходів. Швидкодія залежить від витрат апаратури, тому необхідно проаналізувати апаратурні витрати при різних варіантах організації додаткових переходів між станами автомата в залежності від випадків: наявності умовного переходу між станами автомата, які розглядаються, наявності безумовного переходу і відсутності переходів.

Розглянемо випадки як додати переходи між двох вершин a_i і a_j з додатковим сигналом Sh .

Перший випадок складається в додаванні додаткового сигналу Sh до безумовного переходу (рис. 3.1а). У такому варіанті між вершинами буде не одна дуга з безумовним переходом, а дві дуги з сигналами і Sh (рис. 3.1б).

а)б)

Рисунок 3.1 – Розміщення Sh в разі безумовного переходу

В даному варіанті розміщення функція переходу зі стану a_i в стан a_j матиме вигляд:

Якщо застосувати закон склеювання, то отримаємо a_i . Це означає відсутність додаткових витрат апаратури, якщо буде реалізована мінімальна форма рівняння. При реалізації вихідного рівняння додаткові апаратурні витрати становитимуть один вентиль двовходових (АБО) і два вентиля двовходових (І).

Другий випадок складається в додаванні додаткового сигналу Sh до переходів між станами a_i і a_j , якщо відсутні переходи в інші стани, які відрізняються від стану a_j (рис. 3.2). Це випадок можливий тільки для автомата Мілі, оскільки в автоматі Мура не може бути мультідуг між двома станами.

а) б)

Рисунок 3.2 – Розміщення Sh в разі наявності переходів тільки між a_i і a_j

В даному варіанті розміщення додаткового переходу (рис. 3.2а) функція переходу зі стану a_i в стан a_j матиме вигляд:

Якщо застосувати двічі закон склеювання, то отримаємо a_i . Це означає відсутність додаткових витрат апаратури, якщо буде реалізована мінімальна форма рівняння. Для будь-якого числа кратних дуг між a_i і a_j , результат буде той же. При реалізації вихідного рівняння додаткові апаратурні витрати становитимуть: додаткові входи в усі вентиля, що реалізують терми вихідних рівнянь функції переходів (рис. 3.3а) і один вентиль, який реалізує терм a_iSh (рис. 3.3б).

а) б)

Рисунок 3.3 – Додаткові витрати для другого випадку

Третій випадок складається в додаванні додаткового переходу з сигналом Sh до переходів між станами a_i і a_j , якщо присутні переходи в інші стани, які відрізняються від стану a_j (рис. 3.4).

В даному варіанті розміщення додаткового переходу (рис. 3.4а) функція переходу зі стану a_i в стан a_j і переходу в стан (рис. 3.4б) матиме вигляд:

а) б)

Рисунок 3.4 – Розміщення Sh в разі наявності переходів між станами a_i і a_j та інших переходів з a_i

Якщо застосувати закон Порецького, отримаємо . Це означає, що додаткові апаратурні витрати при реалізації мінімальної форми рівняння для функції переходу в стан a_j – один вентиль, який реалізує терм $a_i Sh$ (рис. 3.3б). При реалізації вихідного рівняння (не мінімального) для функції переходу в a_j додаткові апаратурні становитимуть: один вентиль, який реалізує терм $a_i Sh$ і один додатковий вхід в вентиль, який реалізує терм первинної функції переходів (рис. 3.3а і 3.4б). Якщо число переходів $a_i \rightarrow a_j$ в вихідному графі переходів більше одиниці, то для кожного з цих переходів з'являється додатковий вхід у вентилі.

На додаток до цього, в даному випадку, переходи $a_i \rightarrow a_k$ також дають додаткові апаратурні витрати у вигляді одного додаткового входу в вентиль, який реалізує терм первинної функції переходів. Для прикладу на рис. 3.4б це

Четвертий випадок складається в додаванні додаткового переходу з сигналом Sh між станом a_i і станом a_j , якщо взагалі відсутній перехід $a_i \rightarrow a_j$ (рис. 3.5).

В цьому випадку додаткові апаратурні витрати складають: один вентиль, який реалізує терм $a_i Sh$ і по одному додатковому входу в ті вентилях, які, реалізують, терми первинної функції переходів.

Рисунок 3.5 – Розміщення Sh в разі відсутності переходів між станами a_i і a_j

При цьому, додатковий сигнал Sh не впливає на переходи $a_s \rightarrow a_j$.

Можна зробити наступний висновок – для вибору шляху розміщення додаткових переходів з сигналом Sh з точки зору мінімізації витрат апаратури найкращим є перший варіант, потім другий і найбільш не вигідними є третій або четвертий варіанти.

Для вибору між 3 і 4 способами (при реалізації мінімального рівняння для функції переходів) необхідно визначати число вихідних дуг, які задовольняють умові: , . В якому способі менше таких дуг, такий і треба вибирати, так як додатковий вентиль для реалізації переходу $a_i \rightarrow a_j$ з сигналом Sh має місце в обох випадках.

При реалізації не мінімального рівняння функції переходів в способі 3 при порівнянні зі способом 4 для підрахунку числа дуг до умови, потрібно додати умову крім $a_i \rightarrow a_j$ с Sh .

Додаткові апаратурні витрати, які забезпечують тестопридатність схемної реалізації керуючого автомата при додаванні додаткових переходів, які дають можливість встановити автомат в необхідний стан, залежать від способу кодування станів і складності функцій збудження. Існує пряма залежність між складністю функцій збудження (переходів) і додатковими апаратурними витратами на реалізацію нової дуги Sh в графі переходів

автомата. Тобто, існує можливість зменшити апаратурні витрати при забезпеченні тестопрідатності шляхом знаходження певного порядку розміщення додаткових дуг Sh в графі переходів керуючого автомата.

Для оцінки витрат апаратури реалізації функцій переходів, яка забезпечує переклад КА в необхідний стан a_i , необхідно використовувати показник досяжності.

Коли обираються пари станів автомата, між якими встановлюється додатковий перехід Sh (додаткова дуга у графі переходів), перевагу віддають тому стан-наступнику, для якого сумарна оцінка апаратурних витрат для функцій переходів мінімальна з урахуванням кодів станів автомата. При цьому відсутність переходів між станами вважається рівнозначною переходу з однією умовою.

При обранні переходу $(a_n \rightarrow a_j)$, на якому розміщується додатковий перехід Sh , у разі наявності двох та більше вихідних дуг з вершини a_n , використовується функція переваги

,

де D_n - досяжність вершини, з якої виходить додаткова дуга Sh , D_j - досяжність вершини-наступника, n - кількість вихідних дуг з вершини D_n . Виходячи з цього додаткова дуга ставиться на переході, для якого функція приналежності мінімальна.

Наведений метод розстановки додаткових переходів Sh і обчислення досяжності є евристичним з двох причин. По-перше, на обчислення досяжності впливає спосіб кодування станів автомату, 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, не враховує циклів у графі переходів автомата, що в кінцевому рахунку може вплинути на обчислення досяжності.

3.2 Метод використання зсувного регістру

Ще один спосіб забезпечення тестопридатності КА складається в організації зсувного регістру в послідовних частини автомата. Перед організацією зсувного регістру необхідно виконати автоматизований синтез автомата без зсувного регістру, щоб визначити спосіб кодування станів автомата, запропонований САПР, оскільки від кодування залежить розрядність регістра, що зберігає стани автомата. VHDL модель автомата, заданого ГСА і ГП (рис. 2.1 і 2.2) наведена в додатку Б (Лістинг 1). Після синтезі даної моделі в системі Xilinx ISE Project Navigator з використанням пристрою Xilinx Spartan-3E Starter Kit XC3S500E-5fg320 кодування станів виконано наступним чином (рис 3.6):

```
Analyzing FSM <FSM_0> for best encoding.
Optimizing FSM <State/FSM> on signal <State[1:8]> with one-hot encoding.
```

```
-----
State | Encoding
-----
a0    | 00000001
a1    | 00000010
a2    | 00000100
a3    | 00010000
a4    | 00001000
a5    | 01000000
a6    | 10000000
a7    | 00100000
-----
```

Рисунок 3.6 – Кодування станів автомату

В даному випадку використовувалося унітарне кодування, тому розрядність регістра дорівнює числу станів. Фрагмент VHDL-моделі автомата з зсувним регістром приведена в лістингу 3.1, а повна VHDL-модель наведена в додатку Б (Лістинг 2).

Лістинг 3.1 – Фрагмент FSM зі зсувним регістром

```

-- Блок для формування послідовної частини
Sreg0_CurrentState: process (Clk, reset, State, x1, x2, x3)
begin
    if Reset='1' then State <= "00000000";
    elsif Clk'event and Clk = '1' then
        if A='1' then State <= NextState;
        else State <= TDI & State(7 downto 1);
        end if; end if;
end process;

-- Блок для формування комбінаційної частини
Sreg0_NextState: process (State, x1, x2, x3)
begin
    case State is
        when "00000000"=> NextState <= "00000001";
        when "00000001"=> if x3='1' then NextState <= "00000010";
        ...
        when others => NextState <= "00000000";
    end case;
end process;

-- Опис вихідних сигналів
y1<='1' when State="00000010" else '0';
...
end;
```

Сигнал А - управління режимами регістра, при А = 1 виконується нормальний режим функціонування автомата, при А = 0 виконується зрушення регістра вправо на один розряд. TDI - вхід для послідовного введення інформації в зсувний регістр.

Фрагмент випробувального стенду приведений в лістингу 3.2, а повна VHDL-модель стенду наведена в додатку Б (Лістинг 3).

Лістинг 3.2 - Фрагмент випробувального стенду

```

stim_proc: process
begin
wait for 100 ns;

Reset <='1'; wait for Clk_period;

Reset <='0'; A<='1';wait for Clk_period;

x3 <='1'; wait for Clk_period;
x3 <='0'; wait for Clk_period;
x2 <='1'; wait for Clk_period;

A<='0'; TDI<='1'; wait for Clk_period*8;

wait; end process; END;

```

Часові діаграми роботи автомата зі зсувними регістром наведені на рис.3.7

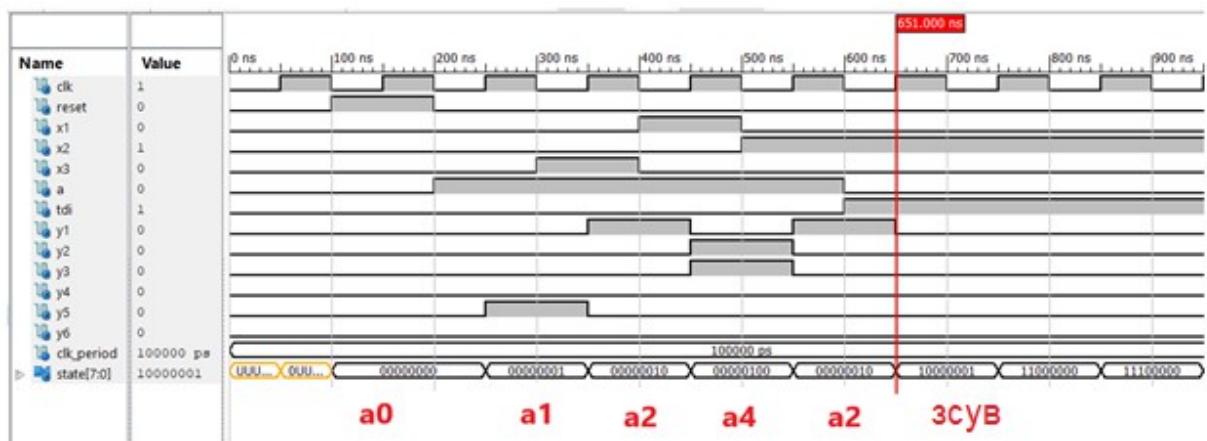


Рисунок 3.7 – Часові діаграми роботи автомата зі зсувними регістром

При використанні зсувного регістру в послідовності частини автомата для установки автомата в заданий стан необхідно завантажити код стану в зсувний регістр з входу TDI. При унітарному кодуванні станів можна організувати циклічний зсув. У цьому випадку досить на тлі нулів зрушувати одиницю в циклічному регістрі зсуву після установки автомата в початковий стан, таким чином переводячи автомат в потрібний стан.

Для діагностичного експерименту в такому випадку недостатньо обійти всі стани. Перекладаючи автомат в потрібний стан, необхідно перевіряти всі переходи зі стану, що завантажується в реєстр. Тоді максимальний час для проведення діагностичного експерименту буде оцінюватися як $T = n^2k$, де n - число станів, а k - максимальне число переходів зі стану. Ще одним недоліком є зростання витрат апаратури (іноді в кілька разів) і зниження швидкодії (іноді до 35%). До переваг цього підходу можна віднести легкотестуємість і мінімальний час побудови тесту, тобто підготовки діагностичного експерименту.

3.3 Порівняльний аналіз запропонованих методів

Для виконання порівняльного аналізу були проведені експерименти над VHDL-моделями. Для вихідної моделі і моделі з зсувними реєстром в послідовних частини експерименти наведені в підрозділі 3.2.

Також були проведені наступні експерименти. Для початку розглянемо VHDL-модель автомата з додатковим сигналом Sh , в якій додаткові дуги з Sh розставлені між станами таким чином, що утворюється гамільтонов цикл. При цьому обрані найкращі пари вершин з точки зору мінімізації витрат апаратури (підрозділ 3.1). В результаті отримано наступний цикл обходу вершин графа переходів: $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_5 \rightarrow a_6 \rightarrow a_4 \rightarrow a_7 \rightarrow a_0$.

Фрагмент VHDL-моделі автомата з сигналом Sh з мінімумом витрат апаратури наведена в лістингу 3.3, а повна VHDL-модель наведена в додатку Б (Лістинг 4).

Лістинг 3.3 - Фрагмент моделі FSM з сигналом Sh з мінімумом витрат апаратури

```
Sreg0_NextState: process (State, x1, x2, x3, Sh)
begin
  case State is
    when a0=> if Sh ='1' then   NextState <= a1;
```

```

else nextState <= a1; end if;
when a1=> if Sh ='1' then    nextState <= a2;
      elsif x3='1' then    nextState <= a2;
      else  nextState <= a1; end if;
...
when a7=> if Sh ='1' then    nextState <= a0;
      else nextState <= a0; end if;
when others => nextState <= a1;
end case; end process;

```

При цьому описі автомата САПР використовувала для кодування код Грея і виділила мінімум тригерів для зберігання станів, а саме три тригера (рис.3.7).

```

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	010
a4	110
a5	111
a6	100
a7	101

Рисунок 3.8 – Кодування станів автомата з *Sh*

Далі розглянемо VHDL-модель автомата з додатковим сигналом *Sh*, в якій додаткові дуги з *Sh* розставлені між станами таким чином, що утворюється гамільтонов цикл. При цьому обрані найгірші пари вершин з точки зору мінімізації витрат апаратури. В результаті отримано наступний цикл обходу вершин графа переходів: $a_0 \rightarrow a_4 \rightarrow a_2 \rightarrow a_7 \rightarrow a_5 \rightarrow a_1 \rightarrow a_3 \rightarrow a_6 \rightarrow a_0$.

Фрагмент VHDL-моделі автомата з сигналом *Sh* з максимумом витрат апаратури наведена в лістингу 3.4, а повна VHDL-модель наведена в додатку Б (Лістинг 5).

Лістинг 3.4 - Фрагмент моделі FSM з сигналом *Sh* з максимумом витрат апаратури

-- Блок для формування комбінаційної частини

Sreg0_NextState: process (State, x1, x2, x3, Sh)

begin

case State is

when a0=> if Sh ='1' then NextState <= a4;

else NextState <= a1; end if;

when a1=> if Sh ='1' then NextState <= a3;

elsif x3='1' then NextState <= a2;

else NextState <= a1; end if;

...

when a7=> if Sh ='1' then NextState <= a5;

else NextState <= a0; end if;

when others => NextState <= a1;

end case; end process;

При цьому описі автомата САПР використовувала для кодування також код Грея і виділила для зберігання станів три тригера (рис.3.8).

Тепер розглянемо VHDL-модель автомата з додатковим сигналом *Sh*, доданим у вигляді єдиного переходу між станами a_6 і a_4 .

При цьому організовується гамільтонов цикл з природних первинних переходів, заданих в автоматі плюс один перехід з *Sh*. В результаті отримано наступний цикл обходу вершин графа переходів:
 $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_5 \rightarrow a_6 \rightarrow a_4 \rightarrow a_7 \rightarrow a_0$.

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    | 110
a3    | 010
a4    | 011
a5    | 100
a6    | 101
a7    | 111
-----

```

Рисунок 3.9 – Кодування станів автомата з *Sh*

Фрагмент VHDL-моделі автомата з сигналом *Sh* з єдиним додатковим переходом $a_6 \rightarrow a_4$ приведена в лістингу 3.5, а повна VHDL-модель наведена в додатку Б (Лістинг 6).

Лістинг 3.5 - Фрагмент моделі FSM з сигналом *Sh* з єдиним додатковим переходом $a_6 \rightarrow a_4$.

```

-- Блок для формування комбінаційної частини
Sreg0_NextState: process (State, x1, x2, x3, Sh)
begin
    case State is
        when a0=> NextState <= a1;
            ...
        when a6=> if Sh = '1' then    NextState <= a4;
                    elsif x2='1' then NextState <= a2;
                    elsif x1='1' then NextState <= a7;
                    else NextState <= a5;    end if;
        when a7=> NextState <= a0;
        when others => NextState <= a0;
    end case; end process;

```

Для даної моделі САПР кодує стани, так само як у вихідному графі переходів (рис. 3.6).

На завершення розглянемо VHDL-модель автомата з додатковим сигналом Sh , доданим у вигляді єдиного переходу між станами a_3 і a_4 , як ми розраховували в розділі 2. При цьому організовується гамільтонов цикл з природних первинних переходів, заданих в автоматі плюс один перехід з Sh . В результаті отримано наступний цикл обходу вершин графа переходів: $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow a_5 \rightarrow a_6 \rightarrow a_7 \rightarrow a_0$.

Фрагмент VHDL-моделі автомата з сигналом Sh з єдиним додатковим переходом $a_3 \rightarrow a_4$ приведена в лістингу 3.6.

Лістинг 3.6 - Фрагмент моделі FSM з сигналом Sh з єдиним додатковим переходом $a_3 \rightarrow a_4$.

-- Блок для формування комбінаційної частини

```
Sreg0_NextState: process (State, x1, x2, x3, Sh)
```

```
begin
```

```
    case State is
```

```
        when a0=> NextState <= a1;
```

```
        ...
```

```
        when a3=> if Sh ='1' then NextState <= a4;
```

```
                elsif x2='1' then NextState <= a2;
```

```
                elsif x1='1' then NextState <= a7;
```

```
                else NextState <= a5; end if;
```

```
        ...
```

```
        when a7=> NextState <= a0;
```

```
        when others => NextState <= a0;
```

```
    end case; end process;
```

У таблиці 3.1 наведені характеристики розглянутих моделей для порівняльного аналізу.

Таблиця 3.1 – Характеристики розглянутих моделей

Характерис-тики	Вихідна FSM	FSM зі зсувним регістром	FSM з Sh (min)	FSM з Sh (max)	FSM з одним Sh $a_6 \rightarrow a_4$	FSM з одним Sh $a_3 \rightarrow a_4$
Тип кодування	Унітарне	Унітарне	Грея	Грея	Унітарне	Унітарне
Швидкодія (max) МГц	362,3	237,1	308,6	275,4	345,1	345,1
Втрати швидкодії, %	-	34	14	24,8	5,24	5,24
Період CLK (min), нс	2,76	4,218	3,24	3,63	2,9	2,9
Кількість Slices	7	15	9	9	6	6
Кількість FF	8	8	3	3	8	8
Кількість LUTs	13	28	15	16	11	11
Кількість IOBs	11	13	12	12	12	12
Витрати по Квайну для КЧ	39	272	108	123	46	42
Загальні витрати по Квайну	143	376	147	165	150	146
Приріст загальних витрат, %	-	162,9	2,79	13,28	4,9	2,1

Спосіб додавання переходу з *Sh* тільки для окремо обраних пар станів для організації гамільтонова циклу спільно з природними переходами між станами автомата, дає мінімальне значення за загальними апаратурним витратам і мінімальне зниження швидкодії. Однак, модифікувати таким способом автомат достатньо складно.

Додавання *Sh* (*Sh min*) між усіма станами для організації гамільтонова циклу дає співмірні значення за загальними апаратурним витратами та зниженням швидкодії, але все ж трохи більші ніж в попередньому випадку. . Однак модифікувати таким способом автомат достатньо просто. І цікаво, що САПР відреагувала на таку модифікацію графа зміною способу кодування станів з унітарного на код Грея, тобто сприйняла нову структуру простіший для оптимізаційних процесів.

При реалізації зсувного регістру спостерігається значне зростання витрат апаратури (на 162,9%) і зниження швидкодії (на 34%) для розглянутого прикладу (для проведених експериментів з більш простими і лінійними ГСА, додаткові апаратурні витрати не перевищували 60%). Таке велике зростання витрат апаратури характерний для САПР ПЛІС. При використанні ASIC технологій для реалізації зсувного регістру необхідно

додати по одному мультиплексу з 2 в 1 для кожного тригера. Це по 7 одиниць за Квайном на кожен тригер. Для розглянутого прикладу автомата це призвело б збільшенню апаратних витрат на 39,8%, а не на 162,9%, як у випадку з ПЛС. Перевагою є те, що час побудови тесту для діагностичного експерименту є мінімальним, но час проведення експерименту зростає.

Залежно від функції мети можна вибрати той чи інший спосіб забезпечення тестопридатності автомата.

ВИСНОВКИ

В результаті проведених досліджень було виконано:

– запропоновано метод аналізу тестопридатності цифрових автоматів, представлених у вигляді орієнтованого графа для спрощення задачі генерації тестів. Метод заснований на топологічному аналізі орієнтованого графа і кодів його вершин;

– удосконалено алгоритм побудови дерева рішень для обходу станів ГП автомата при наявності циклів для організації ДЕ, запропоновано евристичний алгоритм обходу графа за Гамільтоновим циклом;

– розроблено методи підвищення тестопридатності керуючих автоматів, перший метод полягає в організації зсувного регістру для установки автомата в будь-який заданий стан з його допомогою;

– другий розроблений метод полягає в додаванні сигналу Sh , ініціюючого додаткові переходи між станами автомата для організації гамільтонова циклу при обході станів, він мінімізує додаткові апаратурні витрати за рахунок організації визначеного обходу станів автомату.

– зроблено аналіз апаратурних витрат для другого методу при різних варіантах організації додаткового переходу між станами автомату в залежності від наявності безумовного переходу, умовного переходу та відсутності переходів між станами автомату; реалізовано VHDL - моделі для проведення порівняльного аналізу розроблених методів.

Наукова новизна полягає в подальшому розвитку методів оцінки і забезпечення тестопридатності КА та удосконалено алгоритм побудови дерева рішень для обходу станів ГП автомата при наявності циклів для організації ДЕ.

Практична значимість: розроблені методи призначаються для САПР тестопридатних пристроїв та систем, і дають можливість проектувати легко тестовані КА. Наукові дослідження магістерської атестаційної роботи були

апробовані і опубліковані в науковому виданні [48, 49].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Беннеттс Р. Дж. Проектирование тестопригодных логических схем: пер. с англ. / Р. Дж. Беннеттс - М.: Радио и связь; 1990 - 176 с. ил.
2. Abramovichi M. Digital systems testing and testable design / M. Abramovichi, M. A. Breuer, A. D. Friedman, IEEE Inc. New York. 1998. 760 p.
3. Stroud C. E. A Designer's "Guide to Built-in Self-Test" / C. E. Stroud, Kluwer Academic Publishers, 2002. 320 p.
4. Konemann B., Built-in logic block observation techniques / B. Konemann, J. Mucha, G. Swiehoff // Proc. 1979 Test Conf. Oct. 1979, -1979 - p.37-41.
5. Gorodetsky A. Introduction to JTAG and DFT technology. Testing in edge scanning technologies and testable design / A. Gorodetsky Palmarium Academic Publishing, Germany, 2012, 308 p.
6. IEEE Std 1149.1-1990, IEEE Standard for Test Access Port and Boundary-Scan Architecture.
7. IEEE Std 1149.1-2013, IEEE Standard for Test Access Port and Boundary-Scan Architecture.
8. IEEE Std 1500-2005, IEEE Standard Testability Method for Embedded Core-based Integrated Circuits.
9. IEEE Std 1450.6-2006, IEEE Standard Test Interface Language (STIL) for Digital Test Vector Data-Core Test Language (CTL).
10. Network-on-Chip. The Next Generation of System-on-Chip Integration Santanu Kundu, Santanu Chattopadhyay, / CRC Press Taylor & Francis Group, London New York, 2015, 370 p. ISBN: -13: 978-1-4665-6527-2.
11. IEEE Std 1687-2014, IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device.

12. Rutman R.A. Fault Detection Test Generation for Sequential Logic Heuristic Tree Search / R.A. Rutman // IEEE Computer Repository Paper No. R-72-187, 1972.

13. Stephenson J.E. A Testability Measure for Register Transfer Level Digital Circuits / J.E. Stephenson, J.A. Grason // Proc. 6th Fault Tolerant Computing Symposium, 1976, pp. 101 – 107.

14. Grason J. TMEAS - A Testability Measurement Program / J. Grason // Proc. 16th Design Automation Conf., June, 1979, pp. 156-161.

15. Grason J. Digital Test Generation and Design for Testability / J. Grason, A.W. Nagel // Journal Digital Systems, Vol.5, No. 4, 1981, pp. 319-359.

16. Breuer M.A. New Concepts in Automated Testing of Digital Circuits / M.A. Breuer // Proc. EEC Symp. on CAD of Digital Electronic Circuits and Systems, Brussels, 1978, pp. 69-92.

17. Breuer M.A. TEST/80 – a proposal for an advanced automatic test generation system / M.A. Breuer, A.D. Friedman // Proc. IEEE Autotestcon, 1979, pp. 205 – 312.

18. Goldstein L.H. Controllability/Observability Analysis of Digital Circuits / L.H. Goldstein // IEEE Trans. on Circuits and Systems, Vol. CAS-26, No. 9, September, 1979, pp. 685-693.

19. Goldstein L.M. SCOAP: Sandia Controllability/Observability Analysis Program / L.M. Goldstein, E.L. // Proc. 17th Design Automation Conf., June, 1980, pp. 190-196.

20. Berg W.C. COMET: f testability analysis and design modification package / W.C. Berg, R.D. Hess // Proc. IEEE Test Conf., Paper 13.1, 1982, pp. 364-378.

21. Hess R.D. Testability Analysis: an alternative to structured design for testability / R.D. Hess // VLSI Design, March/April, 1982, pp. 22-27.

22. Kovijanic P.G. Computer Aided Testability Analysis / P.G. Kovijanic // Proc. IEEE Automatic Test Conf., 1979, pp. 292-294.

23. Kovijanic P.G. Single Testability Figure of Merit / P.G. Kovijanic // Proc. IEEE. Test Conf., October, 1981, pp. 521-529.

24. Bennetts R.G. CAMELOT: A Computer-Aided Measure for Logic Testability / R.G. Bennetts, C.M. Maunder, G.D. Robinson // IEEE Proc., Vol. 128, Part E, No. 5, 1981, pp. 177-189.

25. Ratiu I.M. Sangiovanni-Vincentelli A., Peterson D.O. VICTOR: A Fast VLSI Testability Analysis Program / I.M. Ratiu, A.Sangiovanni-Vincentelli, D.O. Peterson // Proc. Intn'l Test Conf., November, 1982, pp. 397-401.

26. Dussault J.A. A Testability Measure / J.A. Dussault // Proc. IEEE Semiconductor Test Conf., 1978, pp. 113-116.

27. Spillman R., Glaser N., Peterson D. Development of a general testability figure-of-merit / R. Spillman, N. Glaser, D. Peterson // IEEE International conference of Computer-Aided Design, 1983, pp. 34-35.

28. Chen T-H. Automatic Design for Testability Via Testability Measures / T-H. Chen, M.A. Breuer // IEEE Trans. on Computer-Aided Design, Vol. CAD-4, January, 1985, pp. 3-11.

29. Kulak E. Analysis of the Testability of the High Complexity Digital Devices / E. Kulak, M. Kaminskaya, A. Parfentiy, O. Guz // CADSM'2005, February 23-26, 2005, Lviv-Poliana, pp. 331-333.

30. Кулак Э.Н. Эвристический метод анализа тестопригодности для тестирования цифровых схем детерминированным тестом / Э.Н. Кулак, М. А.Каминская, Ваде Гриби, Хассан Ктейман // Радиоэлектроника и информатика. 2005. №3. С.113-118.

31. Кулак Э.Н. Модификация цифровых схем с использованием метода анализа тестопригодности TADATPG / Э.Н. Кулак, М. А. Каминская, Ваде Гриби, Хассан Ктейман // Радиоэлектроника и информатика. 2005. №4. С.60-68.

32. Kulak E. Testability Analysis Approach TADATPG for Deterministic Test Generation) / E. Kulak, M. Kaminskaya, A. Parfentiy, O. Guz // Electronics and Electrical Engineering 2006. №2 (66), pp. 5-10.

33. Raik J. A New Testability Calculation Method to Guide RTL Test Generation / J. Raik, T. Nommeots, R. Ubar // JOURNAL OF ELECTRONIC TESTING: Theory and Applications 21, 2005, pp. 71–82.

34. Parker K.P. Probabilistic Treatment of General Combinational Networks / K.P. Parker, E.J. McCluskey // IEEE Trans. on Computers, vol. C-24, no. 6, 1975, pp. 668–670.

35. Кулак Э.Н. Анализ тестопригодности цифровых схем на уровне регистровых передач / Э.Н. Кулак, М.А. Каминская, О. Б. Скворцова // Вестник национального технического университета «ХПИ» 2006, №23. С.102-111.

36. Kulak E. Probabilistic Testability Measure Before Pseudorandom Test Generation / E. Kulak, M. Kaminskaya, V. Yeliseev, O. Guz // 13th International Conference “Mixed Design of Integrated Circuits and Systems” MIXDES 2006.– Gdynia. Poland.– 2006.– pp.591-594.

37. Кулак Э.Н. Использование анализа тестопригодности для повышения качества теста и производительности встроенных средств самотестирования / Э.Н. Кулак, М. А. Каминская // Вісник Східноукраїнського національного університету ім. Володимира Даля.– 2008.– №12(130).– Ч.2.– С.24-36.

38. Кулак Э.Н. Метод анализа тестопригодности цифровых устройств для псевдослучайного тестирования в системах встроенного самотестирования / Э.Н. Кулак, М. А. Каминская, Ю.К. Константинова // АСУ приборы автоматки. Вып. 147, 2009. С. 9-14.

39. Kaminskaya M. Hierarchical Analysis of Testability for SoC / M. Kaminskaya, V. Hahanov, E. Kulak, O. Guz // 4th East-West Design and Test Workshop.– Sochi. Russia.– 2006.– pp.226-230.

40. Тоценко В.Г. Алгоритмы технического диагностирования цифровых устройств / В.Г. Тоценко. – М.: Радио и связь, 1985. – 240 с.

41. Богомоллов А.М. Контроль и преобразования дискретных автоматов / А.М. Богомоллов, И.С. Грунский, Д.В. Сперанский. – К.: Наукова думка,

1975. – 175 с.

42. Соколов С.В. Оптимальні та адаптивні системи: навчальний посібник / С.В.Соколов. – Суми: Сум. держ. ун-т, 2018. – 220 с.

43. 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.

44. Kaminska M. Fault Coverage Improving Based on Testability Analysis of the VHDL Code / M. Kaminska, V. Hahanov, A. Hahanova and A. Parfentiy // 2007 9th International Conference - The Experience of Designing and Applications of CAD Systems in Microelectronics, Lviv, UKraine, 2007, pp. 354-356.

45. 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.

46. Шкиль, А.С. Диагностирование hdl-моделей микропрограммных автоматов / А.С. Шкиль, Э.Н. Кулак, А.С. Серокурова // АСУ приборы автоматики. Вып. 172, – 2015. С. 22-31.

47. 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. – pp. 11-13.

48. Трегуб Р.Р. Дослідження впливу заходів щодо забезпечення тестопридатності цифрових керуючих автоматів на апаратурні витрати [Текст] / Р.Р. Трегуб // Тези доповіді 24-го Міжнародного молодіжного форуму «Радіоелектроніка та молодь у 21 столітті» Зб. матеріалів форуму Т.5 – Харків.: ХНУРЕ – 7-9 квітня 2020., с. 9-10.

49. Трегуб Р.Р. Алгоритм построения дерева решений для неразрушающего диагностического эксперимента над конечным управляющим автоматом / Р.Р. Трегуб // Тези доповіді 25-го Міжнародного