

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Т.в.о. завідувача кафедри
_____ Сафонова С.О.
«___» _____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Дослідження методів вбудованого тестування цифрових апаратів

Освітньо-кваліфікаційний рівень “Магістр”
Спеціальність 123 “Комп’ютерна інженерія”

Науковий керівник роботи:

(підпис)

В.С. Кардашук

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я.О. Критська

(ініціали, прізвище)

Студент:

(підпис)

М.С. Руденко

(ініціали, прізвище)

Група:

КІ -18дм

Севєродонецьк 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра комп'ютерних наук та інженерії
Освітньо-кваліфікаційний
рівень “магістр”
Спеціальність 123 – “Комп'ютерна інженерія”
(шифр і назва)
Спеціалізація _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о.зав. кафедри КНІ
к.т.н., доц. С.О. Сафонова
« _____ » _____ 20__ р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Руденко Микита Сергійович

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів вбудованого тестування цифрових апаратів
керівник проекту (роботи) Кардашук В.С., к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
- затверджені наказом вищого навчального закладу від «11» 10 2019 р. № _____
2. Строк подання студентом роботи 10.01.2020
3. Вихідні дані до роботи Матеріали науково-дослідної практики.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____
 1. Огляд методів вбудованого тестування цифрових апаратів
 2. Проектування систем на кристалі та моделювання несправностей в цифрових пристроях
 3. Реалізація вузлів вбудованого тестування та аналіз результатів
 4. Охорона праці та безпека в надзвичайних ситуаціях.
 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основна частина	Кардашук В.С.		
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О.		

7. Дата видачі завдання _____

Керівник _____ Кардашук В. С.

(підпис)

Завдання до виконання прийняв _____ Руденко М.С.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітки
1.	Отримання завдання, збір матеріалів	02.09.19 - 08.09.19	
2.	Огляд літератури й обґрунтування необхідності дослідження	09.09.19 – 13.09.19	
3.	Дослідження існуючих методів реалізації вбудованого тестування	14.09.19 – 14.10.19	
4.	Огляд основних принципів проектування систем на кристалі та дослідження основних несправностей у цифрових пристроях	15.11.19 – 05.12.19	
5.	Дослідження реалізації і роботи граничного тестування методом JTAG	06.12.19 – 04.01.20	
6.	Оформлення пояснювальної записки до дипломної роботи	05.01.20 – 08.01.20	
7.	Підготовка та подання магістерської роботи до захисту	09.01.20 – 10.01.20	

Студент

_____ (підпис)

Руденко М.С.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Кардашук В.С.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Руденко М.С. Дослідження методів вбудованого тестування цифрових апаратів.

Виконано дослідження методів вбудованого тестування цифрових апаратів. Проведено аналіз існуючих варіантів вирішення поставленої задачі та обрано найвдаліший метод для подальшого дослідження. Проведено огляд методів та засобів проектування цифрових схем. Здійснено аналіз типових несправностей у схемах. З метою підвищення якості тесту запропоновано використання технології Boundary Scan. Запропоновано стратегію вибору точок для модифікації цифрової схеми.

Ключові слова: вбудоване тестування, ВІС, НВІС, цифровий апарат, цифрова схема, несправність, регістр.

THE ABSTRACT

Rudenko M.S. Investigation of the methods of integrated testing of digital devices.

Investigation of the methods of integrated testing of digital devices is carried out. The analysis of the existing options for solving the problem is carried out and the most successful method is chosen for further research. An overview of methods and tools for designing digital circuits. Typical faults in the circuits are analyzed. In order to improve the quality of the test, it is proposed to use Boundary Scan technology. A point selection strategy for modifying the digital circuit is proposed.

Key words: integrated testing, LIC, TLIC, digital apparatus, digital circuitry, fault, register.

АННОТАЦИЯ

Руденко Н.С. Исследование методов встроенного тестирования цифровых аппаратов.

Выполнены исследования методов встроенного тестирования цифровых аппаратов. Проведен анализ существующих вариантов решения поставленной задачи и избран самый удачный метод для дальнейшего исследования. Проведен обзор методов и средств проектирования цифровых схем. Осуществлен анализ типичных неисправностей в схемах. С целью повышения качества теста предложено использование технологии Boundary Scan. Предложена стратегия выбора точек для модификации цифровой схемы.

Ключевые слова: встроенное тестирование, БИС, СБИС, цифровой аппарат, цифровая схема, неисправность, регистр.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	8
ВСТУП.....	9
РОЗДІЛ 1	11
ОГЛЯД МЕТОДІВ ВБУДОВАНОГО ТЕСТУВАННЯ ЦИФРОВИХ АПАРАТІВ	11
1.1 Загальні задачі контролю і діагностики цифрових систем.....	11
1.2 Самоперевіряючі цифрові пристрої.....	12
1.3 Методи проектування схем з вбудованими функціями самотестування	13
1.4 Вбудований контроль цифрових схем та систем.....	17
1.5 Особливості вбудованого самотестування мікросхем пам'яті систем на кристалі	20
1.6 Методи вбудованого контролю	23
1.6.1 Технологія ВІЛВО.....	23
1.6.2 Метод периферійного сканування.....	23
1.6.3 JTAG-тестування.....	25
1.7 Генератори тестів для вбудованого тестування дискретних пристроїв	25
1.8 Висновки до розділу 1	27
РОЗДІЛ 2	28
ПРОЕКТУВАННЯ СИСТЕМ НА КРИСТАЛІ ТА	28
МОДЕЛЮВАННЯ НЕСПРАВНОСТЕЙ В ЦИФРОВИХ ПРИСТРОЯХ.....	28
2.1 Розвиток САПР та методології проектування ВІС.....	28
2.1.1 Проблеми побудови САПР.....	28
2.1.2 Вбудовані засоби контролю	28
2.2 Кристал ВІС як системна плата.....	29
2.3 Основні логічні елементи.....	31
2.3.1 Логічні операції та таблиці істинності	32
2.3.2 Параметри інтегральних логічних елементів	33
2.3.3 Класифікація польових транзисторів	36
2.4 Параметри цифрових інтегральних схем.....	36
2.5 Електронні логічні схеми	39
2.6 Логічні схеми транзисторно-транзисторної логіки	41
2.7 Задачі моделювання з несправностями	43
2.8 Типові моделі несправностей	45
2.9 Прояв константних несправностей на функціональному рівні.....	45
2.10 Останні дослідження	46

2.11 Мікроелектронні функціональні цифрові вузли комбінаційного типу	47
2.11.1 Суматори	47
2.11.2 Дешифратори, шифратори, перетворювачі кодів	50
2.11.3 Мультиплектори, демюльтиплектори	52
2.11.4 Цифрові компаратори	54
2.12 Висновки до розділу 2	55
РОЗДІЛ 3	56
ВБУДОВАНЕ ТЕСТУВАННЯ МЕТОДОМ ГРАНИЧНОГО СКАНУВАННЯ JTAG	56
3.1 JTAG-інтерфейс та метод граничного сканування	56
3.2 TAP-порт	58
3.3 Регістри даних	60
3.3.1 Обхідний регістр	61
3.3.2 Регістр ідентифікації МК	61
3.3.3 Регістр даних PDICOM	62
3.4 Механізм граничного сканування	62
3.5 Ланцюг граничного сканування	64
3.5.1 Сканування ліній вводу-виводу	65
3.5.2 Сканування ліній PDI	65
3.6 Трансивер JTAG	66
3.6.1 Дозвіл роботи	66
3.6.2 Відключення	67
3.6.3 Набір інструкцій JTAG	67
3.6.4 Формат посилки та символів	69
3.6.4.1 Спеціальні символи даних	70
3.6.5 Послідовні передача та приймання	70
3.6.6 Послідовна передача	71
3.6.6.1 Сигналізація статусу	71
3.6.7 Послідовний прийом	72
3.6.7.1 Перевірка паритету	72
3.6.7.2 Виявлення символу BREAK	72
3.7 Структурна схема	73
3.8 Місце JTAG-тестування в загальному процесі виробництва	74
3.9 Тестування аналогової частини плати	75
3.10 Інфраструктура JTAG	76

	7
3.11 CLIP: простий логічний аналізатор.....	77
3.12 Зчитування та перевірка зв'язків.....	77
3.13 Мова JTAG Live Script.....	78
3.14 Використання AutoBuzz в Script	79
3.15 Стратегія вибору точок для модифікації комбінованої схеми	80
3.16 Використання технології JTAG.....	81
3.16.1 Тестування	81
3.16.2 Програмування	82
3.16.3 Використання інструментарію.....	83
3.17 Висновки до розділу 3.	83
РОЗДІЛ 4	85
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	85
4.1 Аналіз потенційно небезпечних і шкідливих виробничих факторів, що впливають на персонал.....	85
4.2 Заходи з охорони праці	86
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці.....	87
4.4 Рекомендації з пожежної профілактики	89
4.5 Розрахунок захисного заземлення.....	91
4.7 Перелік джерел посилань до розділу 4	95
ВИСНОВКИ.....	97
ПЕРЕЛІК ПОСИЛАНЬ ДО РОЗДІЛІВ 1-3	99
ДОДАТОК А	101
Приклад роботи в системі JTAG ProVision.....	101
ДОДАТОК Б.....	106
Приклад роботи зі схемою.....	106
ДОДАТОК В	107
Приклад апаратних засобів для JTAG-тестування.....	107
ДОДАТОК Г.....	108
Діаграма станів TAP-контролеру та структура пристрою керування граничного сканування	108
ДОДАТОК Д.....	109
Приклади опису мовами програмування	109
ДОДАТОК Е	111
Електронна презентація	111

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

КС – комп’ютерна система
ТД – технічна діагностика
ОД – об’єкт діагностування
ТС – технічний стан
МОН – метал-оксид-напівпровідник
ВСТ – вбудоване самотестування
КМОН – компліментарна структура метал-оксид-напівпровідник
ЕЗЛ – емітерно-зв’язана логіка
І²Л – інтегрально-інжекційна логіка
МДН – метал-діелектрик-напівпровідник
ІС – інтегральна схема
ВІС – велика інтегральна схема
НВІС – надвелика інтегральна схема
ЦП – цифровий пристрій
ДП – дискретний пристрій
СНК – система на кристалі
ПК – персональний комп’ютер
BSC – Boundary Scan Cells
JTAG – Joint Test Action Group
BILBO – Built-In Logic Block Observer
BIST – Built In Self Test
TAP - Test Access Port
ГС – граничне сканування
ПЗ – програмне забезпечення
ПЗП – постійний запам’ятовуючий пристрій
TAP – Test Access Port
BS – Boundary Scan

ВСТУП

Якість контролю та діагностики залежить не лише від технічних характеристик контрольно-діагностуючої апаратури, але й в першу чергу від тестопридатності самого виробу, що тестується. Це означає, що якість перевірки багато в чому зумовлюється якістю розробки виробів. Найпростіше рішення підвищення якості контролю – це вивід деяких внутрішніх точок виробу на зовнішній роз'єм. Проте кількість вільних контактів на роз'єми обмежена, тому зазначений метод рідко виявляється доступним або достатньо ефективним. Більш прийнятне рішення пов'язано з розміщенням на платі додаткових функціональних елементів, призначених для безпосереднього отримання або накопичення інформації про стан внутрішніх точок та подальшої її передачі на обробку за вимогою аналізу чого пристрою. Сигнали, що з'являються в процесі функціонування основної та контрольної апаратури, що розміщена разом на одному друкованому модулі або кристалі інтегральної схеми, зіставляються за певними правилами. В результаті такого зіставлення отримується інформація про правильне функціонування вузла, що контролюється.

Актуальність теми: Тестопридатне проектування дозволяє спростити процедуру розробки та застосування типових тестів. Використання підходів вбудованого тестування спрямовано на максимальне спрощення роботи тестувальника, тобто дозволяє автоматично виконувати більшість його задач. Використання BIST (Built-in Self-test) – логічне розширення можливостей тестопридатного проектування. Основна задача таких технологій – розмістити генератор тестових впливів та блок оцінки відгуків на тест навколо логіки, що тестується.

Такі технології все частіше застосовуються в останні роки. Їх задача – зменшити вартість тестування шляхом введення критеріїв тестопридатності на самих ранніх стадіях проектування.

Об'єкт дослідження – цифрові пристрої, що підлягають тестуванню.

Предмет дослідження – методи вбудованого тестування систем на кристалі.

Методи дослідження. Для реалізації методів вбудованого тестування цифрових апаратів застосовано моделювання функціональних схем цифрових пристроїв.

Мета дослідження: дослідження методів вбудованого тестування цифрових пристроїв та надання рекомендацій щодо покращення роботи обраного методу.

Наукова новизна магістерського роботи полягає в удосконаленні та подальшому дослідженні методів вбудованого тестування цифрових пристроїв. На основі проведених

досліджень вироблені рекомендації щодо використання комбінацій та поєднань різних методів задля підвищення ефективності вбудованого тестування залежно від потреб.

Апробація результатів роботи. Основні результати роботи представлені у наступних публікаціях:

1. Руденко М.С., Кардашук В.С. Реалізація граничного сканування методом JTAG / Матеріали Всеукраїнської науково-практичної конференції з міжнародною участю «Майбутній науковець-2019» (12 грудня 2019 р.). – Сєверодонецьк. – С. 125-127.

2. Руденко М.С. Дослідження методів вбудованого тестування цифрових апаратів / Збірник науково-практичних праць V молодіжного форуму «ІТ-Ідея 2019» (6 грудня 2019 р.). – Сєверодонецьк. – С. 96-98.

3. Кардашук В.С., Руденко М.С. Дослідження вбудованого тестування цифрових систем / Матеріали ІХ всеукраїнської науково-практичної конференції «Електроніка та телекомунікації» (8-9 листопада 2019 р.) – Сєверодонецьк. – С.121-123.

4. Кардашук В.С., Руденко М.С. Дослідження методів тестопридатності та їх застосування Наукові вісті Далівського університету. Електронне наукове фахове видання. – 2019. - №17.

Практичне використання. Результати дослідження, запропоновані рішення дозволять підвищити якість вбудованого тестування цифрових апаратів та якість і точність самих тестів. Представлені в роботі матеріали, результати досліджень та запропоновані рішення можуть бути використані у навчальному процесі кафедри комп'ютерних наук та інженерії при вивченні таких дисциплін як: «Комп'ютерні системи», «Паралельні та розподілені обчислення», «Цифрова схемотехніка», «Технічна діагностика комп'ютерних систем»

Структура і обсяг роботи.

Магістерська робота складається зі вступу, 4 розділів, висновків, переліку посилань з 44 найменувань на 4 сторінках. Загальний обсяг роботи складає 120 сторінок. Магістерська робота містить 69 рисунків та 13 таблиць.

РОЗДІЛ 1

ОГЛЯД МЕТОДІВ ВБУДОВАНОГО ТЕСТУВАННЯ ЦИФРОВИХ АПАРАТІВ

Одним з найбільш ефективних способів покращення експлуатаційно-технічних характеристик цифрових систем та пристроїв, що зайняли домінуюче положення у сучасних системах, є використання при їх експлуатації методів та способів контролю та технічної діагностики.

Технічна діагностика являє собою область знань, що дозволяє із заданою достовірністю розділяти несправний та справний стани систем і ціль її полягає в локалізації несправностей та у відновленні справного стану системи. З точки зору системного підходу засоби контролю та технічної діагностики доцільно розглядати як складову частину підсистеми технічного обслуговування і ремонту [1-3].

1.1 Загальні задачі контролю і діагностики цифрових систем

Загальні задачі контролю і діагностики цифрових систем та її складових частин зазвичай розглядаються з точки зору основних стадій розробки, виробництва та експлуатації. Поряд із загальними підходами до рішення цих задач маються і суттєві відмінності, що зумовлені специфічними особливостями, що присутні цим стадіям [4-6]. На стадії розробки цифрових систем (ЦС) вирішуються дві задачі контролю та діагностики:

1. Забезпечення контролепридатності цифрової системи в цілому та її складових частин.
2. Налаштування, перевірка справності та працездатності складових частин і системи в цілому.

Системи діагностування призначено для вирішення одної чи декількох задач:

1. перевірки справності;
2. перевірки працездатності;
3. перевірки функціонування;
4. пошуку дефектів.

Технічне діагностування (ТД) здійснюється в системі технічного діагностування (СТД), яка представляє собою сукупність засобів та об'єкта діагностування і, за вимоги, виконувачів, що підготовлена до діагностування та здійснює його за правилами, що зумовлені правилами.

1.2 Самоперевіряючі цифрові пристрої

Метод функціонального діагностування (*on-line*) – метод, при якому наявність у схемі несправностей деякого заданого класу оцінюється під час процесу нормального функціонування схеми. При реалізації даного метода зазвичай стани виходів схеми вважаються спеціальними кодовими словами, що виявляють прояв несправностей з класу, що розглядається. Це досягається введенням додаткових виходів. До виходів схеми підключається додаткова схема вбудованого контролю (детектора кодів), яка виявляє належність вихідних слів, що виробляються схемою, що перевіряється, до кодових слів. Помилка виявляється, якщо вихідне слово, що виробляється системою, не є кодовим словом [7].

Метод потребує додаткового обладнання: детектори, генератори тестових наборів, аналізатори реакцій і т.д. Схеми, в яких додаткове обладнання знаходиться на тому ж чипі, що і схема, яка перевіряється, називаються самоконтрольованими. Самоконтрольовані схеми, реалізовані з використанням методів тестового діагностування, будемо називати самотестуючими, при реалізації функціонального діагностування – самоперевіряючими схемами.

Самоперевіряюча схема має вигляд:

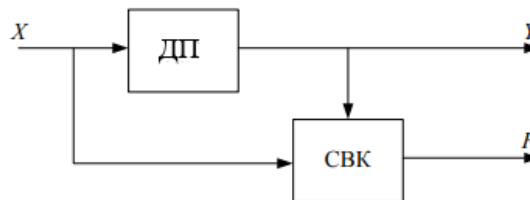


Рисунок 1.1 – Самоперевіряюча схема

Тут ДП – дискретний пристрій, СВК – схема вбудованого контролю. СВК аналізує входи X та виходи Y схеми та видає повідомлення про помилку на виході F .

Будемо називати дискретний пристрій *повністю самоперевіряючим*, якщо будь-яка його несправність із заданого класу виявляється СВК в момент першого її прояву на виходах Y пристрою. Дискретний пристрій будемо називати *частково самоперевірячим*, якщо будь-яка його несправність із заданого класу виявляється схемою вбудованого контролю в робочій області ДП, але не обов’язково в момент першого її прояву на виходах схеми [7].

Якщо СВК виявляє несправності не тільки ДП (об'єкта, що контролюється), але й свої власні, то такі схеми називаються *самоперевіряючими* СВК (ССВК). Самоперевіряючі СВК усувають проблему «сторожа над сторожем», тобто дозволяють виявити несправності в СВК без використання додаткового обладнання.

1.3 Методи проектування схем з вбудованими функціями самотестування

Проблема підлягання тестуванню в надвеликих інтегральних схемах (НВІС) ускладнена нехваткою зовнішніх виводів корпусу мікросхеми для керуємості та спостерігаємості. Методи сканування засновані на поєднанні тригерів, що мають в схемі або спеціально вводяться в неї, в один або декілька зсувних регістрів, що керують станом схеми та керуються через послідовний вхід. Методи самотестування (BIST – Built-In Self-Test) засновані на вбудовуванні в кристал генераторів тестових наборів та схем, що стискають результати перевірки основної частини НВІС за цих тестових наборах.

Найважливішою проблемою в проектуванні НВІС є забезпечення контролепридатності та самотестуючості. Пов'язано це з обмеженою кількістю виводів та неможливістю доступу до всіх вузлових точок НВІС, що проектується. Вимога керуємості на спостерігаємості може бути виконана на основі технології так званого наскрізного зсувного регістру LSSD. Вимога самотестуючості може бути виконана на основі технології ВІЛВО.

Існують дві основні особливості, що характеризують принцип проектування схем на основі метода сканування, чутливих до рівня тактового сигналу (метод LSSD).

Перша полягає в тому, що стани схеми змінюються під керуванням рівня тактового сигналу, а не його фронту. Окрім того, стан, що встановився, виходу як реакція на змінення значень станів первинних входів не залежить від розкиду затримок сигналів при розповсюдженні їх через елементи схеми ті її внутрішні з'єднання. Вихідна реакція схеми не залежить також від порядку переключення вхідних змінних у разі одночасного змінення кількох змінних. Цю властивість «чутливості до рівня» закладено з метою зменшення залежності схеми від її параметрів, що змінюються, таких, як час переключення елементів, затримки розповсюдження сигналів або інші дефекти, які можуть створити умови для перегонів або змагань. Отже, в загальному випадку зменшується вплив можливих фізичних дефектів, які визивають появу несправностей динамічного типу.

Друга особливість методу LSSD полягає в тому, що схема володіє властивістю шляху, що сканується. На рис. 1.2 показано варіант використання зсувно-регістрового

тригера-засувки при реалізації метода LSSD, що називається структурою з одноступеневим тригером-засувкою.

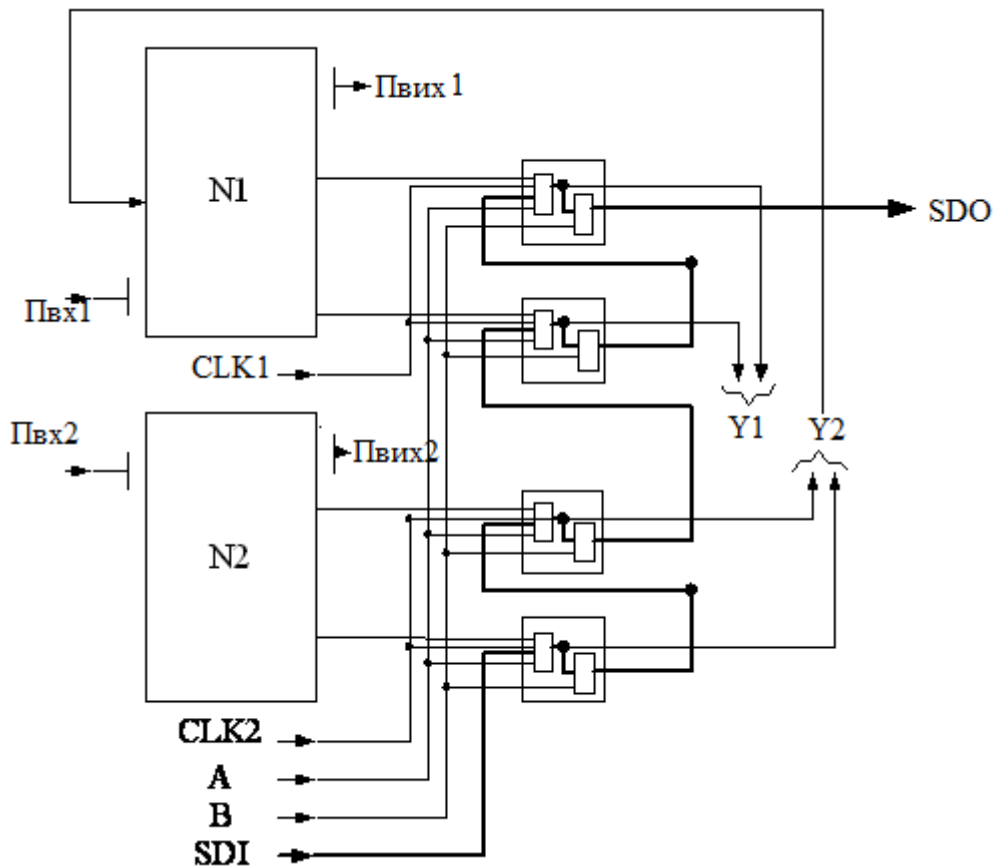


Рисунок 1.2 – варіант використання зсувно-регістрового тригера-засувки

Метод вбудованого поблокового діагностування логічних схем (BILBO – Built-In Logic Block Observation) поєднує елементи сигнатурного аналізу з проектування на основі методології сканування. Розглянемо можливість реалізації безпосередньо в пристрої процедури збору та аналізу даних для того, щоб в ньому забезпечувалася властивість повного самотестування за допомогою вбудованих засобів. Це можна здійснити введенням у пристрій формувачей сигнатур на зсувних регістрах з лінійним зворотнім зв'язком. При цьому потрібно вирішити, де розміщувати ці регістри та скільки їх потрібно мати. Одне з вирішень полягає в використанні таких регістрів для кожної вторинної змінної та зворотному зв'язку, що їй відповідає (рис.1.3); інші - в реалізації для елементів пам'яті схеми додаткової можливості сканування даних та у використанні тільки одного зсувного регістра з зворотним зв'язком (рис.1.4).

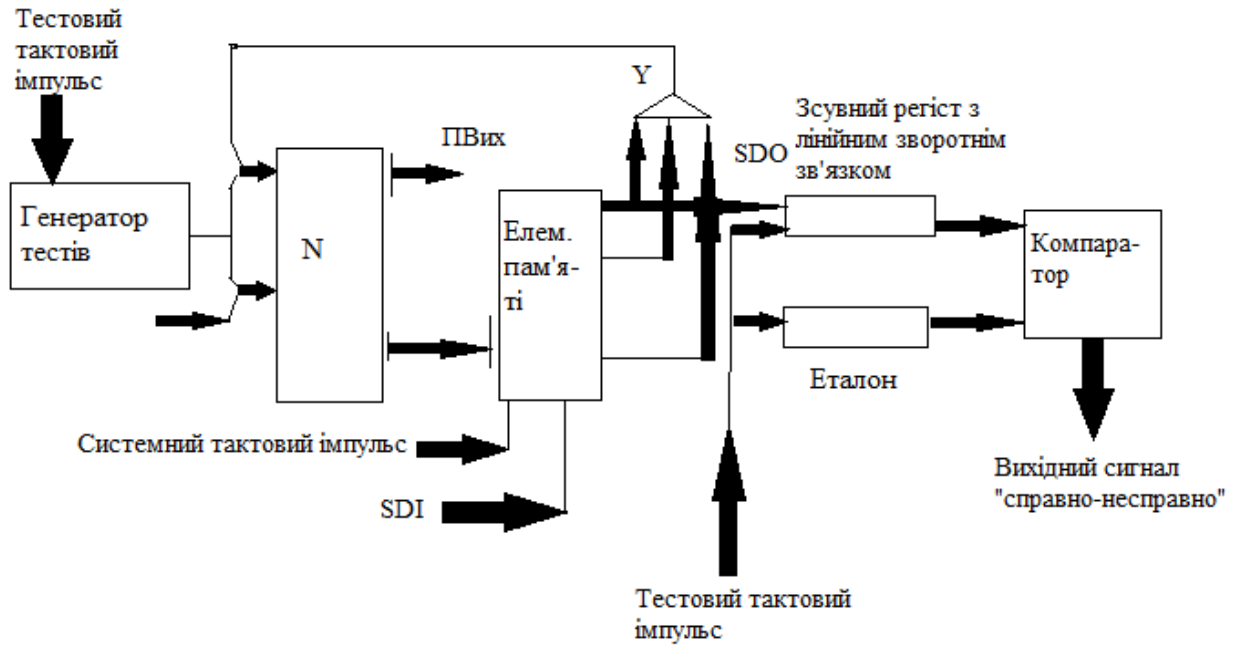


Рисунок 1.3 – Використання зсувних регістрів для кожної вторинної змінної

Тестовий цикл: вибрати режим функціонування, подати в блок N наступний тест, записати реакцію в пам'яті елементи Y, вибрати режим сканування. Ввести дані в зсувний регістр з лінійним зворотним зв'язком. По завершенню зчитати з виходу «справно-несправно» [8].

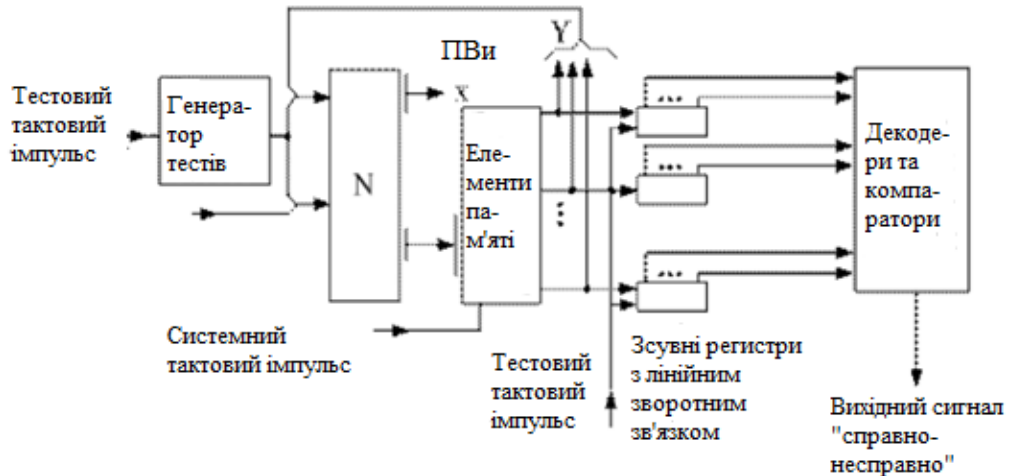


Рисунок 1.4 – Реалізація для елементів пам'яті схеми додаткової можливості сканування даних

Тестовий цикл: прикласти в блок N наступний тест, записати реакцію в пам'яті елемент Y, ввести дані в зсувний регістр з лінійним зворотним зв'язком. По завершенню зчитати з виходу «справно-несправно».

Універсальний елемент VILBO можна використовувати як в схемах з скануванням даних, так і в самотестуючих схемах на основі сигнатурного аналізу. Елемент здатен виконувати декілька різних функцій у відповідності зі значеннями сигналів на двох входах, що керують режимами роботи. Тобто, елемент VILBO являє собою багаторежимний зсувний регістр з додатковими можливостями використання лінійного зворотного зв'язку.

Елемент VILBO можна використовувати у схемах, що реалізують метод сканування LSSD, у якості зсувного регістру в структурі за умови, що прийнято міри для виключення змагань.

На рис.1.5 показано, як можна використовувати елементи VILBO для побудови самотестуючих схем, заснованих на застосуванні сигнатурного аналізу. Для спрощення схеми шини керування, системних тактових імпульсів, скидання та встановлення не наведено [8].

Схема складається з наступних компонентів: V1 – елемент схеми VILBO, що використовується як генератор псевдовипадкових тестових впливів; V2 – інший елемент схеми VILBO, який зазвичай в режимі нормального функціонування є частиною всього пристрою, але в режимі тестування функціонує як багатоканальний аналізатор, що приймає двійкові дані та формує сигнали входних послідовностей.

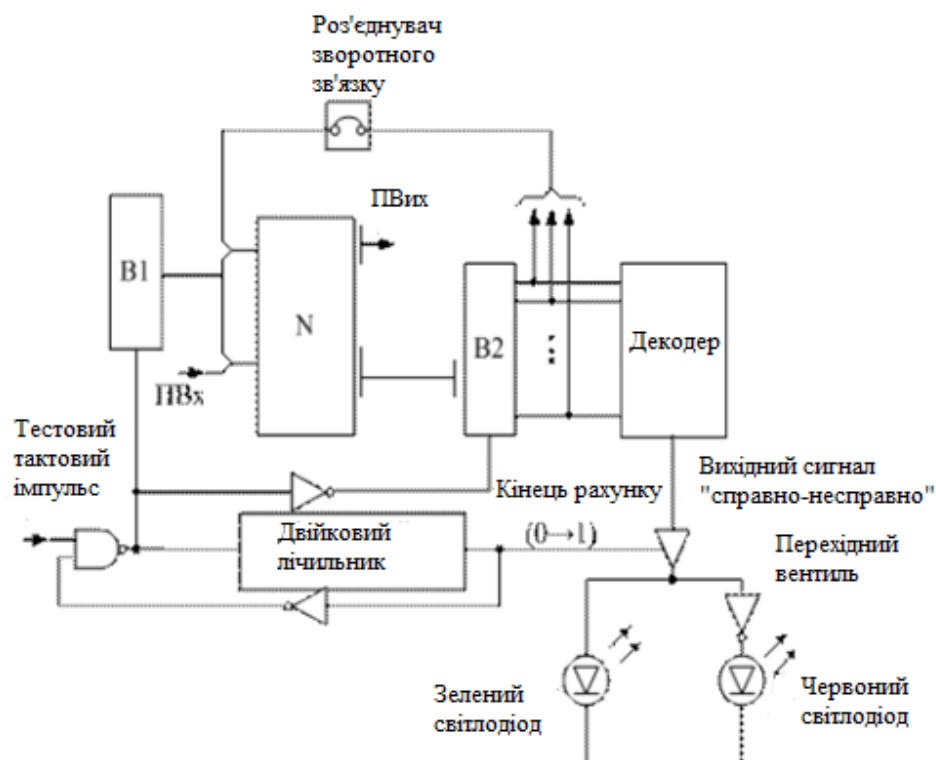


Рисунок 1.5 – Використання елементів VILBO для побудови самотестуючих схем

1.4 Вбудований контроль цифрових схем та систем

Одним з ефективних шляхів вирішення проблеми контролю цифрової системи представляється використання принципу вбудованого контролю, який полягає в тому, що цифрова система і її складові частини розробляються таким чином, щоб забезпечувалася можливість вбудованого контролю без участі будь-якого зовнішнього обладнання. Методи вбудованого контролю можна ієрархічно перерозподілити між різними рівнями від складових частин до цифрової системи в цілому. Вбудований контроль дозволяє проводити перевірку цифрової системи в процесі виконання основних функцій і по суті підвищує експлуатаційну надійність системи, оскільки дозволяє виявляти відмови відразу ж при їх виникненні.

Вбудованим засобам контролю притаманні такі основні переваги:

1. значне скорочення часу відновлення працездатності системи і, відповідно, підвищення загальної експлуатаційної готовності;
2. зменшення чисельності обслуговуючого персоналу, який забезпечує ремонтно-відновлювальні роботи;
3. скорочення видів ремонту і ЗІП за рахунок підвищення достовірності контролю.

Однак необхідно враховувати, що засоби вбудованого оперативного контролю двояко впливають на характеристики контрольованої системи: з одного боку підвищується вірогідність контролю і зменшується час виявлення несправності, з іншого боку, зростає обсяг додаткового обладнання, що в свою чергу призводить до зниження надійності самої системи. Таким чином, вбудовані засоби оперативного контролю, забезпечуючи вигреш в достовірності контролю, призводять до певного програшу в безвідмовності, контрольованої апаратури. У зв'язку з цим пошук розумного оптимуму між повнотою охоплення вбудованим контролем системи і обсягом коштів вбудованого контролю є актуальним завданням. Облік впливу обсягу вбудованого контролю на експлуатаційні характеристики системи дозволить оптимально перерозподіляти ресурси між вбудованими та зовнішніми засобами контролю і діагностики. Тому для обґрунтованого вибору вбудованого контролю необхідне проведення дослідження впливу обсягу коштів вбудованого контролю на такі характеристики, як коефіцієнт готовності, ймовірність виявлення несправності і середній час відновлення цифрової системи.

Існують наступні параметри ефективності системи вбудованого контролю:

1. K_2 – коефіцієнт готовності контрольованої системи з системою вбудованого;
2. $P_{в.тв}$ – ймовірність виявлення несправності контрольним пристроєм;
3. ΔP - прорив в безвідмовності контрольованого пристрою з системою контролю;

4. ΔD – виграш в достовірності при використанні вбудованого контролю;
5. T_o – середній час напрацювання на відмову контрольованої системи з системою вбудованого контролю;
6. T_e – середній час відновлення контрольованої системи з системою вбудованого контролю.

Як зазначено в критерії оцінки ефективності функціонування системи контролю – програш у безвідмовності контрольованого пристрою з системою вбудованого контролю. Він зазначений наступною формулою:

$$\Delta P = |P_{\text{пох}} P_x - P_{\text{пох}}|, \quad (1.1)$$

де $P_{\text{пох}}$ – ймовірність безвідмовної роботи первинної (не контрольованої) схеми;

P – ймовірність безвідмовної роботи.

В свою чергу ймовірність безвідмовної роботи первинної схеми можна зазначити як:

$$P_{\text{пох}} = \frac{1}{1 + \frac{\lambda_o}{\theta}}, \quad (1.2)$$

де λ_o – параметр потоку відмов всього обладнання,

θ – інтенсивність відновлення контрольованої системи.

Об'єктивною тенденцією розвитку сучасних цифрових систем є розширення кола розв'язуваних ними завдань при одночасному підвищенні вимог до ефективності функціонування. Різке збільшення кількості елементів в одиниці обладнання, ускладнення схемних рішень і функціональних зв'язків цифрових систем призводить до значних труднощів в оцінці їх технічного стану, виявлення несправностей і виявленні їх причин в умовах експлуатації. В результаті зростають експлуатаційні витрати, пов'язані з технічним обслуговуванням і ремонтом цифрових систем.

В даний час технологічний процес технічного обслуговування і ремонту цифрових систем не в повній мірі відповідає сучасним вимогам їх експлуатації. Це пояснюється тим, що для виконання технологічних операцій з технічного обслуговування та поточного ремонту, цифрові системи не завжди укомплектовані спеціальними технічними засобами.

Крім того, використовувана при технічному обслуговуванні експлуатаційно-технічна документація не містить рекомендацій щодо виконання технологічних операцій по поточному ремонту і діагностиці відмовили функціональних вузлів (плат) цифровий системи, а обслуговуючий персонал не має достатніх знань, досвіду і навичок в галузі

експлуатації сучасних цифрових систем, створених на базі ВІС, НВІС і мікропроцесорних комплектів.

Однією з основних завдань функціонального контролю в цифрових системах є оперативне виявлення відмов технічних засобів (ТЗ). Для вирішення цього завдання необхідно контролювати стан кожного ТЗ і сам процес передачі та обробки інформації. Контроль процесу в цілому є системним, в більшості випадків він виявляється більш простим в реалізації і досить повним, його елементи включаються в усі протоколи обміну. В існуючих протоколах передачі інформації передбачений контроль вірності інформації, завдяки якому виявляється виникнення будь-якого технічного відмови, що викликає порушення процесу передачі та обробки інформації.

Одним з недоліків контролю процесу в цілому є затримка з виявленням відмови на інтервалі часу від моменту виникнення відмови до його виявлення. З цієї точки зору певні переваги має функціональний контроль стану кожного ТЗ системи, завдяки якому відмова ТЗ може бути заблоковано в момент виникнення відмови. При цьому відмова має виявлятися і усуватися в точці технологічного процесу, найменш віддаленої в часі і просторі від точки виникнення цієї відмови. У більш загальному випадку реальна система функціонального контролю фіксує відмови лише з певною ймовірністю. Відмови, які не фіксуються контролем, виявляються з затримкою часу, яка є в загальному випадку величиною випадковою.

В силу адитивності ця затримка додається до часу відновлення: $t_n = t_b + t_3$, де t_b – випадковий час відновлення, що обчислюється від моменту виявлення відмови до моменту повного відновлення; t_3 – вказаний вище випадковий час затримки виявлення відмови, що обчислюється від моменту, коли фактично відбулася відмова, до моменту його виявлення.

Тому за один з показників якості функціонального контролю ТЗ приймається ймовірність оперативного (т. Е. В момент виникнення або із заданою допустимою затримкою t_3) виявлення відмови $P_{обм}(t)$.

Для забезпечення єдиної стратегії контролю і діагностики цифрових систем доцільне використання двох рівнів: верхній рівень - контроль і діагностика з точністю до ТЕЗа на базі вбудованих засобів контролю, нижній рівень - діагностики несправностей за допомогою засобів технічної діагностики до несправного елемента в Тезе.

1.5 Особливості вбудованого самотестування мікросхем пам'яті систем на кристалі

Інтегральна схема була розроблена в 1958 році, одночасно в Texas Instruments (TI) і Fairchild Semiconductor. Сьогодні напівпровідники лежать в основі багатьох поточних досягнень в галузі електроніки. Впровадження нових технологій, особливо нанометровій технології, дозволило напівпровідниковій промисловості йти в ногу із запитамі споживачів. Витрати на виробництво ІС можуть в даний час складати 40% від загальної вартості продукту. Крім того, якість продукції і її затребуваність можуть значно знизитися, якщо ці чіпи не призначені для тестування [9].

Створення та впровадження системи чипів створює нові проблеми для дослідників. Число вбудованої пам'яті ядра швидко розширюється, і отже займає пам'ять на системних чипах.

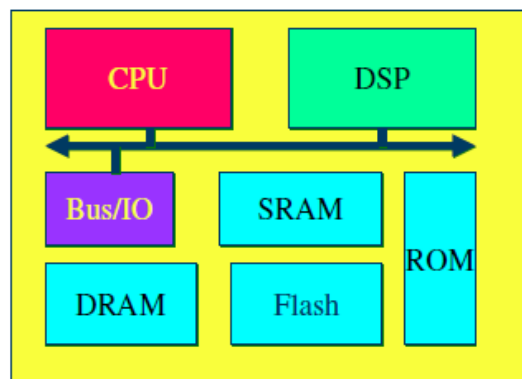


Рисунок 1.6 – Структура ядер систем на кристалі

Таким чином, вихід на чипі пам'яті визначає виходи працездатних чипів. Go / no-go тестування вже не досить для вбудованої пам'яті в системі на чипі (System-on-Chip, SOC). Діагностика пам'яті швидко стає одним з найважливіших питань, оскільки прибутковість виробництва і час виходу на обсяг продукції SOC не стабільна. Застосування методики ефективної діагностики пам'яті і аналіз відмов (FA) дозволяє підвищити якість продуктів SOC.

Розглянемо несправності моделей пам'яті і тестування алгоритмів. Уявімо симулятор несправності пам'яті, званий випадковий симулятор доступу до пам'яті, для помилки скринінгу, яка складається з моделювання численних дескрипторів. Моделювання зчитує тестові впливу і встановлює прапори операцій для кожної комірки пам'яті. Несправність покриття визначається шляхом перевірки несправності дескрипторів для певних умов. Генератор моделювання та випадкових алгоритмів буде представлений, на

основі тест-алгоритмів RAMSES і March. March-тести вважаються найбільш ефективними для звичайних моделей оперативної пам'яті. Їх легко створювати і, як правило, вони короткі [10].

З появою глибоких субмікронних з дуже високим ступенем інтеграції (VLSI) технологій, продавці звертаються до SOC. Тому що майже всі чипи системи містять деякі види вбудованої пам'яті, пам'ять вважають одним з найбільш універсальних компонентів. Відсоток від вартості вбудовується пам'яті на загальному ринку напівпровідникової пам'яті виросте до 50% і більше в майбутньому. Так як ця тенденція широко поширена, тестування вбудованої пам'яті отримує все більшу увагу з боку промисловості, а також наукової спільноти.

Є багато проблем у об'єднанні пам'яті (DRAM, флеш-пам'ять і т. Д.) з логікою. На додаток до існуючих технічних проблем для гарантування продуктивності, якості та надійності вбудованої пам'яті ядра, потрібні подальші дослідження.

У методології дизайн-для-перевірки (Design-For-Test) повинна бути передбачена ізоляція ядра тестом доступу, який повинен бути включений в собівартість розробки, але це в свою чергу знижує продуктивність і викликає небажані ефекти. Навіть якщо ця методологія є здійсненним, то тестування пам'яті для повної кваліфікації та тестування вбудованої DRAM (EDRAM) буде більш дорогим, через збільшення швидкості введення/виведення і обсягу даних [13]. Перспективним рішенням цієї дилеми є використання технології BIST (рис. 1.7).

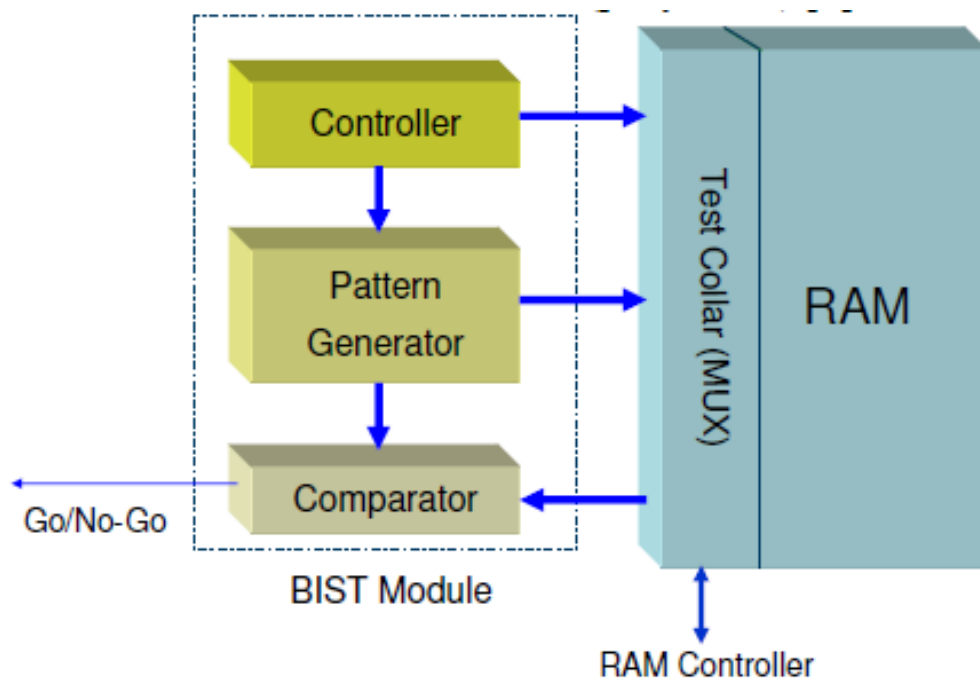


Рисунок 1.7 – Типова архітектура пам'яті BIST

Із застосуванням BIST вимога тестування EDRAM може бути зведена до мінімуму, а час тестування пам'яті може бути зменшено протягом усього терміну перевірки EDRAM. Крім того, загальний час тестування може бути зменшено, оскільки паралельне тестування в банку пам'яті виконувати простіше [12].

Таким чином, BIST буде, широко застосовуватися для тестування EDRAM. Ще одна перевага BIST полягає в тому, що це хороший підхід до захисту інтелектуальної власності (IP), тобто виробник (EDRAM ядра в даному випадку) повинен поставляти тільки активації BIST і послідовності реагування для тестування і діагностики цілей без розкриття деталей проектування [11].

Хоча BIST був успішно застосований для вбудованих SRAM (ESRAM), його успіх у вбудованих DRAM, флеш-пам'яті, CAM і т.д., ще належить з'ясувати. Необхідність зовнішнього тестування пам'яті EDRAM не може бути проігнорована, якщо аналіз надмірності і ремонт можна зробити на чіпі. Крім того, нові види відмов або збоїв повинні бути протестовані і за допомогою March алгоритмів, таких як ті, які використовуються в схемах ESRAM BIST, але які вважаються недостатніми.

Існують інші проблеми. Поєднання вбудованого датчика пам'яті і BIST логіки також є цікавою темою. Наступним завданням є автоматизація проектування. Ще однією проблемою є синхронізація кваліфікації тестування або змінної пам'яті від асинхронної пам'яті з синхронної BIST логікою.

Таким чином, йдеться про важливі аспекти тестування напівпровідникової пам'яті, в тому числі, тестування алгоритмів, моделювання несправностей, автоматичну генерацію алгоритмів тестування і самодіагностики. Архітектура BIST підтримує March -тести і діагностику. Вибравши відповідний підхід, вона також перевіряє терміни специфікацій. Підхід є гнучким, оскільки додаткові команди тесту (крім елементів March) можуть бути включені з мінімальними витратами. Використання архітектури BIST є економічно ефективним, так як випробування займає мінімум часу, витрати на апаратні засоби низькі, і тестове покриття є об'ємним. Компілятор BIST для вбудованої пам'яті називається мозком, який може бути використаний для загальних SRAM і DRAM ядер, таких як: синхронні SRAM, асинхронні SRAM, подвійний порт SRAM, два порти реєстрового файлу, ZBT SRAM, EDO DRAM, SDRAM, DDR DRAM і т. д. З огляду на пам'ять специфікації і вимоги до випробувань, мозок генерує синтезується (RTL) код замикання для BIST в Verilog, а також активацію його послідовностей, випробувальний стенд, і синтез сценаріїв. BIST впливає на швидкість тестування і діагностики оперативної пам'яті тестованого, а March-тести є програмованими. Такий інструмент може бути використаний для широкого спектра оперативної пам'яті, архітектур і конфігурацій. Крім того, BIST-схеми можуть бути

загальними для декількох видів пам'яті, щоб знизити загальні накладні області. Тому використання BIST дуже важливо для проектування SOC.

1.6 Методи вбудованого контролю

1.6.1 Технологія BILBO

Реалізація функції самотестування може бути виконана на основі технології BILBO (Built-In Logic Block Observer). Метод вбудованого поблочного діагностування логічних схем BILBO поєднує елементи сигнатурного аналізу з проектуванням на основі методології сканування. Тут передбачена можливість реалізації процедури збору та аналізу даних безпосередньо в пристрої для того, щоб в ньому забезпечувалась властивість повного самотестування за допомогою вбудованих засобів. Це можливо здійснити введенням у пристрій формуваачей сигнатур на зсувних регістрах з лінійним зворотнім зв'язком. Ідея реалізована у вигляді універсального елемента, що забезпечую властивість до самотестування. Такий універсальний елемент може виконувати всі функції генератора тестових послідовностей, формувача сигнатур, навіть запом'ятовуючого пристрою зі скануванням даних. Цей елемент може використовуватись як у схемах зі скануванням шляху, так і в самотестуючих схемах на основі сигнатурного аналізу. Елемент BILBO можна використовувати у схемах, що реалізують метод сканування LSSD, в якості зсувного регістра в структурі за умови прийняття мір для виключення змагань.

1.6.2 Метод периферійного сканування

Спочатку цей метод було призначено для перевірки з'єднувальних ланцюгів на друкованих платах та HBIC, при цьому зсувні регістри вводяться на кристал, обслуговуючи кожен зовнішній вивід. При перевірці міжз'єднань відключаються ланцюги всередині кристалу, а для перевірки IC підключаються регістри, що знаходяться всередині, а також генератори тестових послідовностей та схеми стиснення на основі сигнатурного аналізу. Підключення пристрою до тестера виконується за допомоги п'яти ліній. Для проектування схем із периферійним скануванням розроблено стандарт IEEE 1149.1 та спеціальні мови BSDL (Boundary Scan Description Language) та HSDL (Hierarchical Scan Description Language), які є мовами опису апаратури в ряду VHDL. На рис. 1.8 показано з'єднання елементів в рамках однієї плати, що підтримує даний стандарт.

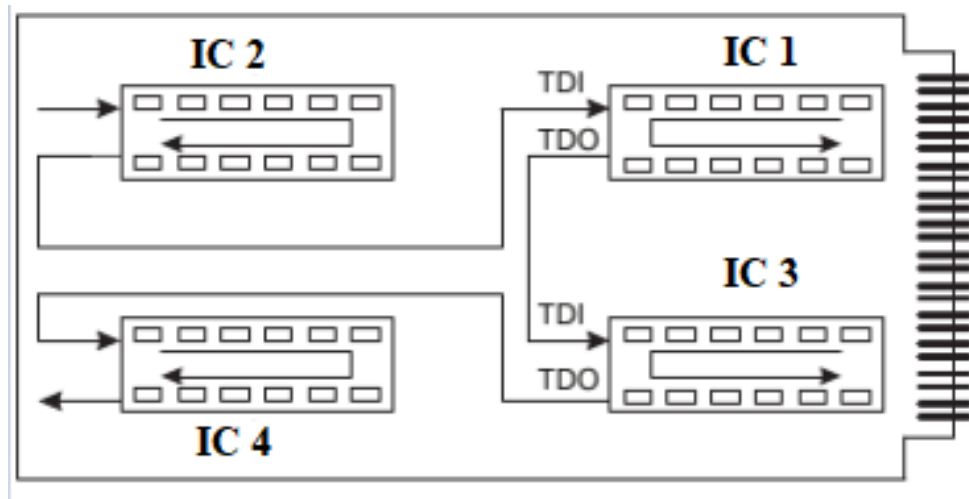


Рисунок. 1.8 – З'єднання елементів периферійного сканування в структурі однієї плати

Можливий варіант периферійного сканування на рівні системи представлено на рис. 1.9. Тут вихід даних попередньої плати з'єднується з інформаційним входом наступної системної плати.

Основна ідея периферійного сканування полягає в тому, що послідовний регістр зсуву розташовується по периферії мікросхем, а осередки цього регістру знаходяться між зовнішніми виводами та функціональною частиною інтегральних мікросхем. У відповідності до стандарту до кожної інтегральної схеми повинні додаватися п'ять зовнішніх контактних виводів: TDI (Test data In – вхід тестових даних), TMS (Test Model Select – вибір тестового режиму), TCK (Test Clock – тестова синхронізація), TRST (асинхронне скидання).

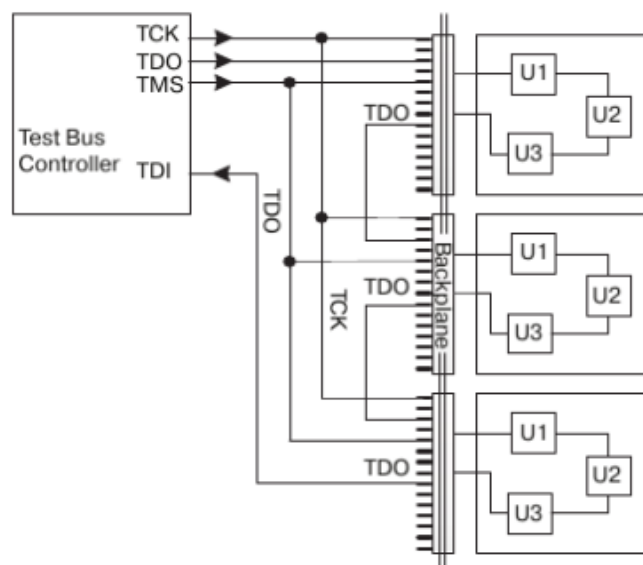


Рисунок 1.9 – Структура підключення на рівні системи

Ці методи та стандарти, а також стандарт ГС ІЕЕЕ 1532, що забезпечує конфігурування всередині схеми, є головним напрямком застосування технологій периферійного сканування, що пов'язане з усуненням несправностей прототипів та тестуванням якості монтажу, збірки вузлів та систем.

1.6.3 JTAG-тестування

Ідея тестування плати через JTAG-інтерфейс – це внутрісистемне тестування. Суть метода полягає в тестуванні плати через спеціальний невеликий роз'єм за допомоги 4-провідного інтерфейсу, що закріплений стандартом ІЕЕЕ 1149.1 ще в 1990 році. Цей інтерфейс часто ще називають JTAG-інтерфейсом. Маючи у складі пристрою одну або декілька мікросхем, що підтримують стандарт ІЕЕЕ 1149.1, можна протестувати не тільки ланцюги, пов'язані з цими мікросхемами, але й інші елементи, включаючи пам'ять, логіку, резистори та зовнішні роз'єми. При переході в режим граничного сканування JTAG-компоненти відключають свою основну логіку та переходять в режим тестування, дозволяючи зовнішньому обладнанню керувати своїми выводами та тестувати ланцюги.

Інструменти, призначені для тестування по JTAG-інтерфейсу, дозволяють також зробити внутрісистемне програмування Flash-пам'яті і ПЛІС. Величезна кількість мікросхем, що використовується в сучасних розробках, підтримують стандарт ІЕЕЕ 1149.1 - це вимога сучасної електронної індустрії. Досить згадати серед них сигнальні процесори Texas Instruments, ПЛІС від Altera, Xilinx і Lattice, процесори Qualcomm і багато інших. Цілком можливо, що ваша розробка вже готова до граничного сканування, так як виробники мікросхем вже заклали в них надлишкову тестову логіку.

1.7 Генератори тестів для вбудованого тестування дискретних пристроїв

Вбудоване тестування дискретних пристроїв (ДП) широко використовується у виготовленні сучасних ДП на рівні друкованих плат та отримує подальший розвиток при проектуванні та створенні систем на одному кристалі (СОК). Підвищення швидкодії логічних елементів, складності СОК та необхідність перевірки їхньої справності на робочих частотах ДП, що досягають 1/5 ГГц, зумовлює переваги вбудованого тестування перед зовнішніми засобами діагностування. На сьогоднішній день ведучими світовими спеціалістами було висунуто ідею суміщення концепції BIST та стандарту проектування ІЕЕЕ 1149.1 «Периферійне сканування». Цей стандарт визначає структуру СОК, в якій вхідні регістри функціонують в двох режимах: робочому та тестовому. В тестовому режимі

вхідні регістри СОК реконфігуруються в зсувні регістри (СР), що дозволяє вводити та виводити діагностичну інформацію через стандартний порт JTAG та звести процедуру діагностування до перевірки справності комбінаційної частини ДП. В режимі самотестування вхідні регістри реконфігуруються в генератори тестів та еталонних сигнатур, а вихідні регістри в синдромно-сигнатурні аналізатори вихідних реакцій ДП, що перевіряється. В якості генераторів тестів використовуються зсувні регістри з лінійними та нелінійними зворотними зв'язками (ЗРЛЗЗ та ЗРНЗЗ), мережі кліткових автоматів (МКА) [16].

Генератори послідовностей з перетворювачами тестових векторів. Широке застосування методів компактного тестування ДП шляхом використання генераторів псевдовипадкових та псевдовичерпних тестів з наступним стисненням вихідної реакції синдромно-сигнатурними аналізаторами в якості вбудованих засобів діагностування СОК зумовило низку нових проблем при проектуванні СОК та їхньому діагностичному забезпеченню. Перша проблема пов'язана з необхідністю підвищення ступеню покриття несправностей СОК щонайменше для класу константних несправностей, друга – зумовлена необхідністю зниження вживаної енергії в процесі тестового діагностування (ТД).

Довжина псевдовичерпних тестів для СОК з кількістю входів $50 \div 200$ може скласти $10^4 \div 10^6$ тестових наборів. При цьому не виявляються несправності вузлів, які є важкокерованими, схем, що перевіряються [17]. Крім того, так як споживана енергія в КМОН схемах пропорційна числу перемикачів її елементів, то скорочення довжини тестових послідовностей дозволить скоротити час тестування і мінімізувати енергетичні витрати на технічне обслуговування СОК.

Одним з підходів для вирішення цієї проблеми є метод тестування, заснований на використанні генераторів тестів на основі ЗРЛЗЗ і перетворювачів послідовностей, що генеруються, які з окремих тестових наборів формують безліч детермінованих тестів (ДТ), що виявляють цільові несправності в певних вузлах схеми (рис.1.10). При цьому довжина тестової послідовності не змінюється. Використання безлічі ДТ, що покривають 100% константних несправностей, є найбільш економічним рішенням завдання тестування СОК за умови реалізації схем генерації тестів з мінімальними апаратними витратами.

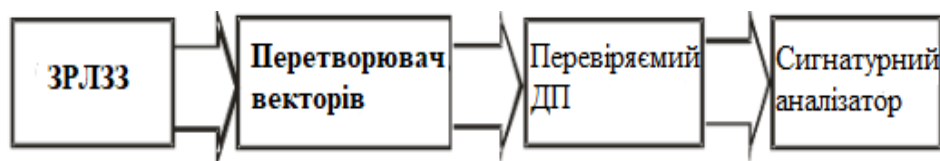


Рисунок 1.10 – Метод тестового діагностування на основі ЗРЛЗЗ та перетворювача тестових векторів

Одним з рішень цього завдання є використання ПЗП, вбудованого на кристал або друковану плату, для запису і зберігання повної множини ДТ, синтезованих системами генерації тестів і моделювання несправностей на етапі проектування СОК. Управління адресними лічильниками ПЗУ дозволяє формувати тести для послідовних схем шляхом вибірки сканованих тестових наборів, які здійснюють установку тригерів схеми, яка перевіряється, і тестових наборів комбінаційної частини схеми, які докладають до схеми, яка перевіряється, по закінченню сканування. Мінімізація обсягу ПЗП для зберігання ДТ здійснюється шляхом використання методів розбиття безлічі тестів на підмножини, комбінації яких дозволяють скоротити обсяг пам'яті. При цьому ускладнюється структура схеми управління процесом діагностування, що не враховується в більшості відомих робіт [18, 19].

1.8 Висновки до розділу 1

В першому розділі розглянуто загальні задачі та проблеми тестопридатного проектування на вбудованого тестування і діагностики цифрових пристроїв. Розглянуто основні методи реалізації вбудованого тестування. Серед розглянутих методів для подальшого дослідження обрано метод граничного сканування, а саме його реалізацію за допомогою методу JTAG. Таке рішення було прийняте через універсальність та багатофункціональність методу JTAG. Завдяки стандарту IEEE 1149.1 та його модифікаціям ми отримуємо можливість:

- реалізації вбудованого тестування;
- реалізації вбудованого самотестування;
- програмування схеми;
- налаштування ВІС.

Отже, обраний метод поєднує в собі кращі властивості своїх конкурентів, хоча його основним недоліком і є залежність від програмного забезпечення та обладнання для тестування. Незважаючи на це, саме метод Boundary Scan.

РОЗДІЛ 2

ПРОЕКТУВАННЯ СИСТЕМ НА КРИСТАЛІ ТА МОДЕЛЮВАННЯ НЕСПРАВНОСТЕЙ В ЦИФРОВИХ ПРИСТРОЯХ

2.1 Розвиток САПР та методології проектування ВІС

2.1.1 Проблеми побудови САПР

Згідно до «закону Мура» кількість транзисторів на кристалі зростає експоненціально в часі. З ускладненням інтегральних пристроїв зростає кількість вимог до їх елементів, що призводить до над експоненціального зростання необхідних обчислень в системі проектування.

Сучасна технологія напівпровідникових мікросхем забезпечує створення виробів електронної техніки, що поєднують на одному кристалі більше мільярда транзисторів. При використанні традиційних методів проектування для таких ультра ВІС потрібні великі трудові та обчислювальні ресурси, а також нові програмні засоби проектування та дуже великі фінансові затрати. Вже на рівні складності у декілька мільйонів транзисторів проявляються обмеження традиційної методології проектування, в якій мікросхема розглядається як один об'єкт [15].

Перша проблема – це проблема великих чисел, тобто зростання об'ємів обчислень.

Друга проблема – це ускладнення фізичних процесів (і їх розрахунків) для нанометрових розмірів елементів. Наприклад, квантові явлення та статичний розкид кількості легуючих атомів в активних областях транзисторів, дискретний характер електричних зарядів при малих токах.

Третя проблема – людський фактор. Розробникам доводиться створювати все більш об'ємні специфікації на проекти. Дуже важко оцінити оптимальність отриманих рішень. Великі об'єми обчислень створюють і великі масиви результатів. Потребуються більш високі рівні опису проектів та подання результатів для керування процесом розробки.

2.1.2 Вбудовані засоби контролю

Контрольно-вимірального обладнання для тестування ВІС в реальному масштабі часу зараз немає і створити його неможливо. Сучасні поєднання важкості та швидкості досяжні тільки в складі ВІС. Контроль сигналів можна здійснити тільки спеціальними

реєструючими блоками, що розміщені на тому ж самому кристалі. Проектування системи вбудованого контролю повинно здійснюватись одночасно з розробкою основної системи.

2.2 Кристал ВІС як системна плата

Основна ідея в розвитку методології проектування замовних – це використання в проекті вже готових функціонально завершених блоків. Приблизно так, як це робиться при розробці системи на платі з використанням універсальних мікросхем.

Системи на кристалі (СНК) – це методологія розробки замовних мікросхем на основі вже готових складно функціональних блоків (СФ-блоків). Основою методології є сумісність СФ-блоків в системі по принципу «ввімки та працєю» (plug-and-play).

Для реалізації цього принципу СФ-блоки повинні розроблятися як автономні пристрої зі своїми системами харчування, синхронізації і інтерфейсами.

Перша складова методології систем на кристалі - це єдині вимоги до СФ-блокам. Обов'язковими є вимоги технологічної сумісності, наявність детальних специфікацій і моделей високого рівня.

Друга складова - це конкретні базові рішення, забезпечують виконання вимог сумісності СФ-блоків.

СФ-блоки повинні включати і елементи інфраструктури системи (Інтерфейси, системи харчування і синхронізації, вбудовані засоби контролю). Інфраструктурні блоки не повинні займати велику площу кристала і використовувати багато зовнішніх компонентів.

Третя складова - це універсальні правила, що забезпечують об'єднання СФ-блоків в систему з найменшим взаємним впливом.

Маршрут проектування СНК істотно скорочується і спрощується в порівнянні з маршрутом повністю замовлених мікросхем. Методологія проектування СНК наближається до методології розробки систем на друкованих платах. Основний етап проектування - це системний. Саме на цьому етапі визначаються всі основні характеристики мікроелектронного пристрою, що розроблюється. Етапи функціонального проектування і верифікації об'єднуються і спрощуються. Моделювання схеми на транзисторному і вентильному рівнях взагалі може не проводитися. Використовуються тільки моделі високого рівня. Можливо і виключення етапу макетування СНК, якщо всі використовувані СФ-блоки атестовані і адекватно описані на мовах високого рівня (VHDL, VHDL-AMS і ін.). Фізичне проектування також істотно спрощується, тому що число використовуваних СФ-блоків і сигнальних зв'язків між ними порівняно невелика. По суті, СНК є напівзамовними мікросхемами і основні витрати припадають на створення системи

проектування і поширення СФ-блоків. Основна вигода полягає в тому, що кожен СФ-блок використовується в багатьох виробках. Крім цього, в кілька разів скорочується час розробки кінцевих продуктів.

Методологія проектування систем на кристалі наказує виконання проекту за двома напрямками.

Напрямок «згори - донизу» включає:

- складання загальної специфікації на СНК;
- розробку системної моделі;
- підготовку номенклатури СФ-блоків, що потребується;
- функціональне моделювання СНК;
- фізичне проектування;
- верифікацію моделі.

Напрямок «знизу - вгору» включає:

- підготовку специфікації на потрібні СФ-блоки;
- відбір готових блоків;
- придбання або розробку відсутніх блоків;
- розробку та верифікацію моделей високого рівня для СФ-блоків, що використовуються.

За рівнем витрат на розробку і підготовку виробництва СНК займають проміжне місце між універсальними мікросхемами і ПЛІС. Проміжних рівнів можна виділити кілька. Самий витратний рівень - це комплектування проекту наявними СФ-блоками і розробка відсутніх. При цьому потрібно повний цикл фізичного проектування кристала. У структурних СНК на базовому кристалі вже розміщені СФ-блоки. Функціональна схема формується із заданого набору СФ-блоків шляхом створення системи металізованих сполук. Якщо структура СФ-блоків на базовому кристалі повторює структуру осередків ПЛІС, то проект можна повністю налагодити на макеті з ПЛІС, а потім перенести на базовий кристал. Такі СНК називають "жорсткі копії ПЛІС" (FPGA Hard Copy). Виграш досягається за рахунок виключення системи програмування з'єднань. Площа кристала при цьому скорочується до 10 разів, відповідно підвищується швидкодія і знижується споживана потужність. найдешевший спосіб розробки – це СНК, що конфігуруються. По суті - це вже структурні ПЛІС. Розробник програмує і функції СФ-блоків і зв'язки між ними. Відмінність від регулярних ПЛІС полягає в тому, що СФ-блоки спеціалізовані та досить різноманітні. Спеціалізація блоків дозволяє в кілька разів скоротити площу кристала в порівнянні з регулярними ПЛІС.

2.3 Основні логічні елементи

Логічний елемент - це така схема, у якій кілька входів і один вихід. Кожному стану сигналів на входах, відповідає певний сигнал на виході.

Польовий транзистор - транзистор, в якому сила струму, що проходить крізь нього, регулюється зовнішнім електричним полем, тобто напругою. Це принципова відмінність між ним та біполярним транзистором, де сила основного струму регулюється керуючим струмом.

Оскільки у польового транзистора немає керуючого струму, то у нього дуже високий вхідний опір, що досягає сотень Гіга і навіть терагерц (проти сотень кіло у біполярного транзистора).

Польові транзистори іноді називають уніполярними, оскільки носіями електричного заряду в ньому виступають тільки електрони або тільки дірки.

У роботі ж біполярного транзистора, як випливає з назви, бере участь одночасно два типи носіїв заряду - і електрони і дірки.

У наш час в логічних схемах використовуються МДН-транзистори з діелектриком SiO_2 (МОН-транзистори). Аналіз МОН-транзисторних логічних елементів достатньо простий, так як через відсутність вхідних струмів їх можна розглядати окремо від інших елементів навіть при роботі в ланцюгу. На рисунку 2.1 показано два варіанта побудови логічних елементів на МОН-транзисторах з n-каналами.

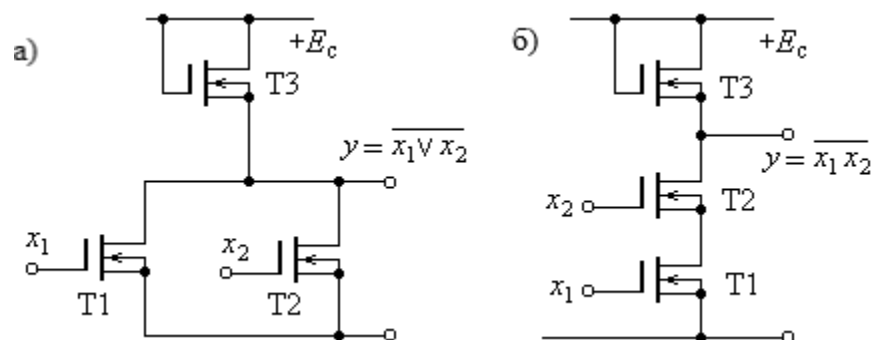


Рисунок 2.1 – Логічні елементи на МОН-транзисторах: а) – елемент АБО-НІ; б) – елемент І-НЕ

Транзистори T3 виконують роль навантаження. Логічні рини в обох схемах не залежать від навантаження та відповідають вихідним напругам відкритого та закритого ключа:

$$U_{\text{вих}}^0 \approx 0,1\text{В}, U_{\text{вих}}^1 \approx E_c \quad (2.1)$$

Відповідно, логічний перепад складає:

$$U_{\text{л}} = U_{\text{вих}}^1 - U_{\text{вих}}^0 \approx E_c \quad (2.2)$$

Напругу живлення E_c МОН-логіки обирають в 3-4 рази більшу порогової напруги U_0 відкриття транзисторів. Якщо $U_0 = 1,5 \dots 3\text{В}$, то логічний перепад, що отримуємо, в $5 \dots 10\text{В}$ набагато перевищує значення, що властиві схемам І²Л, ЕЗЛ та навіть ТТЛ (за напруги живлення $4 \dots 5\text{В}$). Тому МОН-логіка має підвищену завадостійкість. Більш високою швидкістю та низьким енергоспоживанням характеризується логіка на компліментарних транзисторах. За принципом дії та схемотехніки КМОН-логіка дуже близька до МОН-логіки.

2.3.1 Логічні операції та таблиці істинності

Кон'юнктор (елемент І) – реалізує операцію «логічне множення». Схема має два або більше входів та один вихід. На виході сигнал «1» з'являється тоді і тільки тоді, коли на всі виходи одночасно впливають вхідні сигнали «1».

Позначення: $F=A \& B$.

Таблиця 2.1 – Таблиця істинності для кон'юнкції

A	B	F
1	1	1
1	0	0
0	1	0
0	0	0

Диз'юнкція – це складний логічний вираз, який є істинним, якщо хоча б один з простих виразів істинний і помилковий тоді і тільки тоді, коли обидва простих вирази помилкові. Диз'юнкція – це логічне складання.

Позначення: $F = A \cup B$.

Таблиця 2.2 – Таблиця істинності для диз'юнкції

A	B	F
1	1	1
1	0	1
0	1	1
0	0	0

Імплікація – це складний логічний вираз, який істинний в усіх випадках, окрім якщо із істини слідує помилка. Тобто дана логічна операція зв’язує два простих логічних вирази, з яких перше є умовою (А), а друге (В) є наслідком.

« $A \rightarrow B$ » істинно, якщо з А може слідувати В.

Позначення: $F = A \rightarrow B$.

Таблиця 2.3 – Таблиця істинності для імплікації

A	B	F
1	1	1
1	0	0
0	1	1
0	0	1

Логічна рівнозначність. Еквівалентність – це складний логічний вираз, який є істинним тоді і тільки тоді, коли обидва простих логічних вирази мають однакову істинність.

« $A \leftrightarrow B$ » істинно тоді і тільки тоді, коли А і В рівні.

Позначення: $F = A \leftrightarrow B$.

Таблиця 2.4 – Таблиця істинності для еквівалентності

A	B	F
1	1	1
1	0	0
0	1	0
0	0	1

Операція XOR (виключаючі або). « $A \oplus B$ » істинно тоді, коли істинно А або В, але не обидва одночасно. Цю операцію також називають «складання за модулем два».

Позначення: $F = A \oplus B$.

Таблиця 2.5 – Таблиця істинності для виключаючих або

A	B	F
1	1	0
1	0	1
0	1	1
0	0	0

2.3.2 Параметри інтегральних логічних елементів

Незалежно від належності до тієї чи іншої серії, всі логічні елементи характеризуються певним одним і тим же набором параметрів, котрі є довідниковими даними. Значення цих параметрів зумовлені схемо технічним конструктивним та технологічним виконанням елементів.

Значення параметрів, як правило, задаються з запасом та не вичерпують фізичні можливості мікросхеми, однак перевищувати їх не треба.

Оцінюють мікросхеми за наступними параметрами:

1. швидкодія;
2. напруга живлення;
3. вживана потужність;
4. коефіцієнт розгалуження по виходу;
5. коефіцієнт об'єднання по входу;
6. завадостійкість;
7. енергії переключення;
8. надійність;
9. стійкість до кліматичних впливів.

Основні з них розглянемо нижче:

1. Рівні вихідних напруг.

Технічними умовами для кожної серії логічних елементів задаються найбільший та найменший рівні вихідних напруг, що відповідають логічним одиниці та нулю за допустимих змін напруги живлення, навантаження, температури. Напруга $U_{вих\ min}^1$ відповідає мінімальному рівню логічної одиниці на виході (для ТТЛ $U_{вих\ min}^1 = 2,4В$), а напруга $U_{вих\ max}^0$ – максимальному рівню логічного нуля (для ТТЛ $U_{вих\ max}^0 = 0,4В$).

2. Статична завадостійкість.

Цей параметр визнає припустиму напругу на входах мікросхеми та оцінюється для низького та високого рівнів напруги. Статичною завадостійкістю по низькому рівню вважають різницю:

$$U_{зав}^0 = |U_{вих\ max}^0 - U_{вх\ max}^0|, \quad (2.3)$$

Де $U_{вих\ max}^0$ – максимальна припустима напруга низького рівня на виході навантаженої мікросхеми; $U_{вх\ max}^0$ - максимальна припустима напруга низького рівня на вході мікросхеми, що навантажує.

Завадостійкість по високому рівню визначають так:

$$U_{зав}^1 = |U_{вих\ min}^1 - U_{вх\ min}^1|, \quad (2.4)$$

тут $U_{\text{вих min}}^1$ - мінімальна напруга високого рівня на виході навантаженої мікросхеми;
 $U_{\text{вх min}}^1$ - мінімальна припустима напруга високого рівня на вході, що навантажує.

1. Коефіцієнт розгалуження по виходу.

Цей параметр Краз (навантажувальна здатність) визначає максимальну кількість входів елементів даної серії, якою можна навантажувати мікросхеми без порушення їх нормального функціонування.

2. Коефіцієнт об'єднання по входу.

$K_{\text{об}}$ визначає кількість логічних входів, які має логічний елемент. Найпростіші логічні елементи випускаються з 2,3,4 та 8 входами. Більш складні пристрої мають і інші входи: адресні, установлюючі, дозволяючі, входи синхронізації та інші.

3. Вхідні струми.

Ці параметри визначають навантаження, яке представляє схема, що розглядається, на попередню схему або інше джерело живлення. Розрізняють вхідні струми $I_{\text{вх}}^0$ та $I_{\text{вх}}^1$ при подачі логічних нуля або одиниці.

4. Середня статична вживана потужність.

Визначається наступним чином:

$$P_{\text{ст.ср.}} = \frac{1}{2} (P_{\text{вж}}^0 + P_{\text{вж}}^1) \quad (2.5)$$

Де $P_{\text{вж}}^0$ та $P_{\text{вж}}^1$ – потужності, що вживаються інтегральним логічним елементом у стані логічного нуля та логічної одиниці. Це витікає з того, що в складних багатоелементних пристроях в середньому половина логічних елементів знаходиться у стані 1, а половина – у стані 0.

5. Швидкодія.

Характеризується максимальною частотою зміни вхідних сигналів, за якої ще не порушується нормальне функціонування пристрою.

Інерційність напівпровідникових приладів та паразитні ємності слугують причиною того, що кожне переключення супроводжується перехідними процесами, завдяки чому фронти імпульсів розтягуються.

Для оцінки часових властивостей мікросхем зазвичай користуються затримкою розповсюдження сигналу, яка являє собою інтервал часу між вхідним та вихідним імпульсами, що міряються на рівні 0,5. Затримки розповсюдження сигналу при ввімкненні $t_{\text{зт.р.}}^{1,0}$ та при вимкненні $t_{\text{зт.р.}}^{0,1}$ не є однаковими, тому користуються усередненим параметром $t_{\text{зт.р.ср.}} = 0,5(t_{\text{зт.р.}}^{1,0} + t_{\text{зт.р.}}^{0,1})$.

Для послідовних пристроїв (тригери, лічильники і т.д.) вводяться деякі додаткові часові параметри, зумовлені принципом дії: дозволяючий час, тривалість вхідного імпульсу та інші.

В загальному випадку аналіз фізичних, технологічних та схемо технічних особливостей інтегральних логічних елементів показує, що можна створити різні їх варіанти, але їхньою особливістю буде або відносно висока (висока) швидкодія за низької економічності, або висока економічність при відносно низькій (низькій) швидкодії.

2.3.3 Класифікація польових транзисторів

Польові транзистори (FET: Field-Effect-Transistors) поділяються на два типи:

- з керуючим PN-переходом (JFET: Junction-FET);
- з ізольованим затвором (MOSFET: Metal-Oxid-Semiconductor-FET).

Кожен з типів може бути як з N-каналом, так і з P-каналом. В ролі носіїв електричного заряду виступають:

- у транзисторів з N-каналом - електрони.
- у транзисторів з P-каналом - дірки.

2.4 Параметри цифрових інтегральних схем

Логічні та схемотехнічні можливості базових функціональних елементів визначаються сукупністю електричних та функціональних параметрів, які є основними для мікросхем. Основні параметри є спільними для всіх існуючих та можливих логічних інтегральних мікросхем (ІМС) і дозволяють порівнювати між собою мікросхеми різних типів. До цих параметрів відносяться:

- логічна функція, що реалізується;
- швидкодія;
- коефіцієнт об'єднання по входу;
- коефіцієнт розгалуження по входу (навантажувальна здатність);
- завадостійкість;
- споживана потужність;
- стійкість проти зовнішніх впливів;
- ступінь інтеграції, надійність.

Швидкодія ІС визначається середнім часом затримки (t_{zm}) сигналу. Вона визначає час проходження через одну мікросхему у пристрої. При визначенні середньої затримки в

якості меж часових інтервалів зазвичай беруть точки на фронтах, що відповідають половині перепаду напруги, або точки, що відповідають рівням 0,1 ті 0,9 цього перепаду. Якщо ланцюг складається із N послідовно ввімкнених однотипних логічних ІМС, то час проходження сигналу по ланцюгу однієї ІС:

$$T = \frac{N}{2} t_{зт.ввім.} + \frac{N}{2} t_{зт.вимк.} \quad (2.6)$$

тоді середній час затримки визначається як:

$$t_{зт.ср.} = \frac{t_{зт.ввім.} + t_{зт.вимк.}}{2} \quad (2.7)$$

та залежить від режиму роботи транзистора в мікросхемі та споживаної потужності.

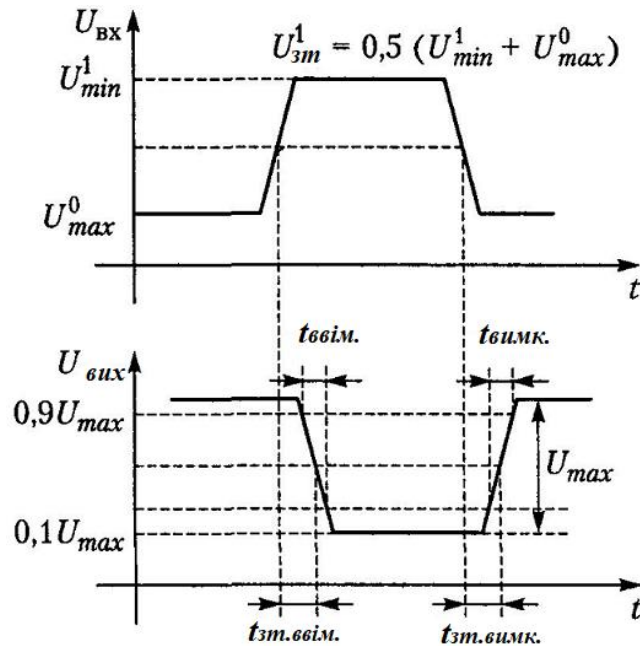


Рисунок 2.2 – Схема проходження сигналу через інтегральну логічну схему

За середнім часом затримки ІС поділяються на:

1. надшвидкодійні, $t_{зт.ср.} < 5$ нс;
2. швидкодійні, 10 нс $> t_{зт.ср.} > 5$ нс ;
3. середньої швидкодії, 100 нс $> t_{зт.ср.} > 10$ нс;
4. низької швидкодії, $t_{зт.ср.} > 100$ нс.

Схеми ТТЛ-типу відносяться до схем середньої швидкодії. Найбільшу швидкодію мають транзисторні логічні схеми з емітерними зв'язками: $t_{зт.ср.} = 1-10$ нс.

Коефіцієнт об'єднання по входу m – це максимальна кількість входів, що може мати логічний елемент. Із його збільшенням розширюються можливості мікросхеми за рахунок виконання більш складних функцій на одному типовому елементі. Збільшення коефіцієнта об'єднання по входу погіршує інші параметри мікросхеми: швидкодію, завадостійкість та навантажувальну здатність. Частіше за все він не перевищує 8, що визначається обмеженою кількістю виводів ІС. Для збільшення m в ІМС вводять спеціальну схему (логічний розширник), підключення якої до основного елемента дозволяє збільшити його до 10 і більше.

Коефіцієнт розгалуження по виходу n (навантажувальна здатність) визначається кількістю схем цієї ж серії, входи яких можуть бути приєднані до виходу даної схеми без порушення її працездатності. Чим вище n , тим ширші логічні здатності мікросхеми і тим менше таких мікросхем необхідно для побудови складного обчислювального пристрою. Однак, зі збільшенням цього коефіцієнта погіршуються **завадостійкість** та **швидкодія**.

Навантажувальна здатність ІС в значному ступені визначається типом інвертора, що в них використовується. Для найпростішого інвертора, що складається із одного транзистора, $n = 2-4$, для складних – $n = 10-20$.

В схемах на основі МДН-транзисторів входи наступних схем в статичному режимі практично не навантажують виходи попередніх. Це дає можливість отримати дуже великий коефіцієнт розгалуження по виходу. Однак в динамічному режимі ємності приєднаних входів сповільнюють перехідні процеси та збільшують струм, що споживається даною схемою.

Завадостійкість $U_n \max$ – це найбільше значення напруги на вході мікросхеми, за якого ще не відбувається змін рівнів вихідної напруги. Завадостійкість визначає працездатність логічного елемента за наявності різних перешкод, що діють на вході ІМС поряд із корисним сигналом. Перешкоди можуть як з'являтися в самих логічних схемах, так і наводитися від сторонніх пристроїв.

Поміхи бувають **статичні** та **динамічні**. Під **статичними** розуміють поміхи, тривалість котрих значно перевершує тривалість перехідних процесів в логічних елементах.

До **імпульсних (динамічних)** завад відносяться короткострокові імпульси, тривалість яких порівнянна з тривалістю перехідних процесів в логічних елементах.

Статична завадостійкість – це найменша постійна напруга, яка, будучи доданою (за самих не благодійних обставинах) до корисного вхідного сигналу, зміщає робочу точку на передаточній характеристиці в область переключення, що визиває помилкове спрацювання по всьому наступному ланцюгу логічних схем. Логічна ІМС в статичному режимі може знаходитись в одному із двох станів – **відкритому** або **закритому**. Тому

розрізняють завадостійкість закритої схеми по відношенню до завад, що відмикають, та завадостійкість відкритої схеми по відношенню до замикаючих.

За статичною завадостійкістю логічні елементи можна умовно розділити на елементи:

1. з низькою завадостійкістю, $U_{a\text{ ст}} = 0,2-0,4\text{ В}$;
2. із середньою завадостійкістю, $U_{n\text{ ст}} = 0,4-0,8\text{ В}$;
3. з високою завадостійкістю, $U_{a\text{ ст}} > 0,8\text{ В}$.

Імпульсна завадостійкість завжди вище статичної. Це викликано тим, що при короткому імпульсі поміх паразитні ємності в логічному елементі не встигають перезарядитися до порогових рівнів переключення ІМС. Тому за однакової статичної завадостійкості схеми з меншим часом затримки більше уражені дії імпульсних перешкод.

2.5 Електронні логічні схеми

Електронною логічною схемою називають схему, в якій вихідний електричний сигнал пов'язаний із вхідним за законами алгебри логіки.

Оскільки в алгебрі логіки змінні можуть приймати тільки два значення («0» та «1»), то всі електричні сигнали в електронних логічних (ЕЛС) схемах мають тільки два значення, що різко відрізняються, одне з яких відповідає логічному нулю, а інше – логічній одиниці.

В найпростішому випадку електричними сигналами цифри «0» та «1» можуть бути представлені двома способами: потенційним та імпульсним.

При потенційних сигналах значенням «0» та «1» відповідають різні рівні напруги $U^{(0)}$ та $U^{(1)}$ або струму $I^{(0)}$ та $I^{(1)}$. Якщо за «1» приймають більш високий рівень напруги (струму), то кажуть про позитивну логіку (рис. 2.3, а). Якщо за «1» приймають більш низький рівень напруги (струму), то кажуть про негативну логіку (рис. 2.3, б).

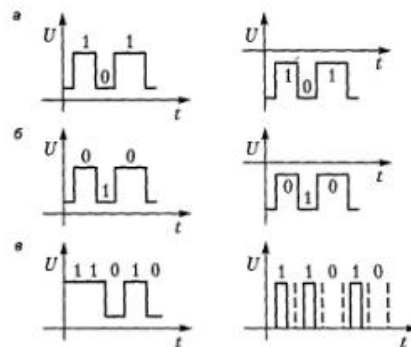


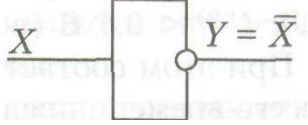
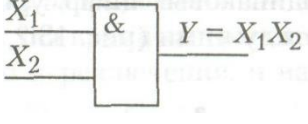
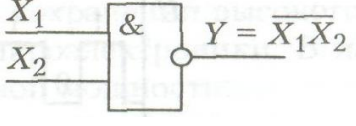
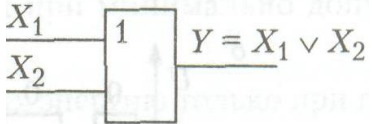
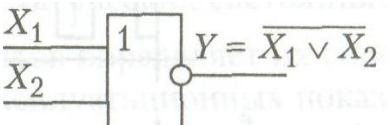
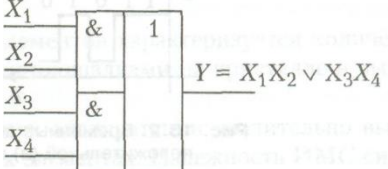
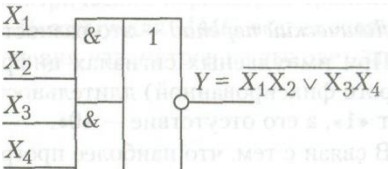
Рисунок 2.3 – Часові діаграми, що пояснюють принцип позитивної (а) та негативної (б) логіки.

Логічний перепад - це різниця між рівнями «0» та «1». При імпульсних сигналах цифри виражаються імпульсами відповідної (суворо фіксованої) тривалості. Наприклад, наявність імпульсу відповідає «1», а його відсутність – «0». У зв'язку з тим, що найбільш прогресивними з конструктивної точки зору є інтегральні схеми, а за інтегральною технологією неможливо виготовляти трансформатори ті важко конденсатори великої ємності, то обирають такі сигнали, при впливі яких нема необхідності в цих елементах. Ними є потенційні сигнали, тому в теперішній час найбільш розповсюджені логічні інтегральні схеми, призначені для роботи з потенційними сигналами. Електронні логічні схеми реалізуються на різній елементній базі.

Так як в інтегральній мікроелектроніці найбільш технологічні резистори, діоди, біполярні та польові транзистори, саме ці елементи використовуються в логічних схемах.

В таблиці 2.6 наведено основні логічні функції, позначення елементів та їх схеми.

Таблиця 2.6 – Основні логічні функції, позначення та схеми елементів

Елемент	Позначення	Схема
НЕ	ЛН	
I	ЛІ	
I-НЕ	ЛА	
АБО	ЛЛ	
АБО-НЕ	ЛЕ	
I-АБО	ЛС	
I-АБО-НЕ	ЛР	

2.6 Логічні схеми транзисторно-транзисторної логіки

Розробка технології виготовлення багатомірного транзистора призвела до створення транзисторно-транзисторних логічних (ТТЛ) схем. Найпростіша схема базового елемента ТТЛ, що реалізує операцію I-НЕ, показана на рис. 2.4, а. В ТТЛ-елементі роль вхідних діодів грають емітерні переходи, а роль діода зміщення - колекторний перехід транзистора VT₁. Транзистор VT₂ разом з резистором R_к являють собою інвертор.

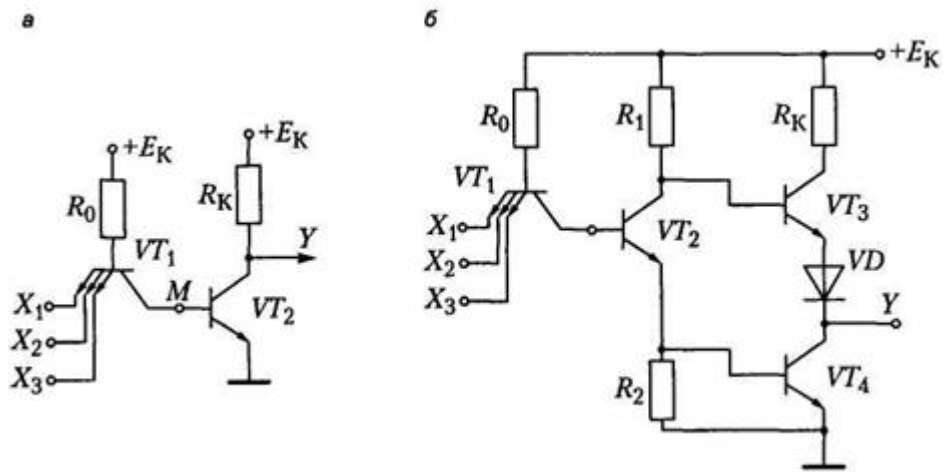


Рисунок 2.4 – Схема базового елемента ТТЛ, що реалізує операцію І-НЕ: а – із простим інвертором; б – зі складним інвертором

Крім того, багатоемітерний транзистор VT_1 займає на кристалі менше місця, ніж діоди в ДТЛ - елементі, тому ТТЛ - елементи більш зручні в інтегральних схемах.

Розглянемо принцип дії схеми (рисунок 2.4, б), що реалізує логічну операцію І-НЕ ТТЛ. Транзистори VT_1 та VT_2 - схема І-НЕ, а транзистори VT_3 , VT_4 – неінвертуючий вихідний каскад, призначений для посилення потужності вихідного сигналу.

1. Нехай на входи X_1 , X_2 , X_3 надходить напруга, рівна напрузі логічної одиниці. Всі емітерні переходи транзистора VT_1 , зміщені у зворотному напрямку і не пропускають струм. Колекторний перехід транзистора VT_1 включений прямо.

Через резистор R_0 і колекторний перехід VT_1 від джерела живлення $+E_K$ в базу транзистора VT_2 надходить струм, достатній для насичення транзистора (напруга на колекторі транзистора VT_2 близька до нуля). Має місце «О».

Отже, при «О» на колекторі VT_2 , який знаходиться у відкритому стані (режим насичення), частина його емітерного струму втікає в базу транзистора VT_4 і насичує його, тобто транзистор VT_4 відкритий, а транзистор VT_3 закритий, так як на колекторі транзистора VT_2 рівень напруги відповідає напрузі логічного нуля.

Таким чином, на виході логічного елемента отримуємо сигнал з рівнем логічного нуля, оскільки на відкритому транзисторі VT_4 є мале падіння напруги.

2. Подамо на один з входів X_1 , X_2 , X_3 напругу, рівну напрузі логічного нуля. Один з емітерних переходів VT_1 зміститься в прямому напрямку. Струм від джерела живлення $+E_K$ через резистор R_0 кинеться до вхідного ланцюга, що володіє меншим опором, ніж вхідний опір транзистора VT_2 .

В результаті струм бази транзистора V_{T2} буде прагнути до нуля, отже, транзистор V_{T2} закривається, а на його колекторі встановлюється високий потенціал напруги, близький до потенціалу джерела живлення $+E_K$ («1»).

Такий стан транзистора V_{T2} (на колекторі транзистора V_{T2} рівень «1», він замкнений) є причиною припинення надходження струму бази в транзистор V_{T4} , і він теж закривається. Висока напруга на колекторі транзистора V_{T2} викликає насичення транзистора V_{T3} . В результаті на виході логічного елемента з'являється «1».

2.7 Задачі моделювання з несправностями

Моделювання цифрових пристроїв з несправностями є одним з найважливіших розділів логічного моделювання та використовується в системах автоматизованого проектування та діагностики діагностуючих пристроїв (ДП) при вирішенні наступних задач:

- визначення ефективності тестової послідовності – її повноти та діагностичних властивостей;
- побудова діагностичних словників для пошуку несправностей в логічних схемах;
- генерація перевіряючих тестів;
- аналіз поведінки схеми та її властивостей з несправністю.

Наявність несправності може суттєво змінити поведінку схеми та призвести до наступних наслідків:

- несправність може визвати змагання сигналів;
- несправність може визвати осциляцію (нескінченну зміну сигналів);
- несправність може заважати встановлення схеми с пам'яттю в початковий стан;
- несправність може перетворити комбінаційну схему в послідовну або синхронну в асинхронну та т.д.

Процес моделювання несправних схем цифрових схем представлено на рис. 2.5.

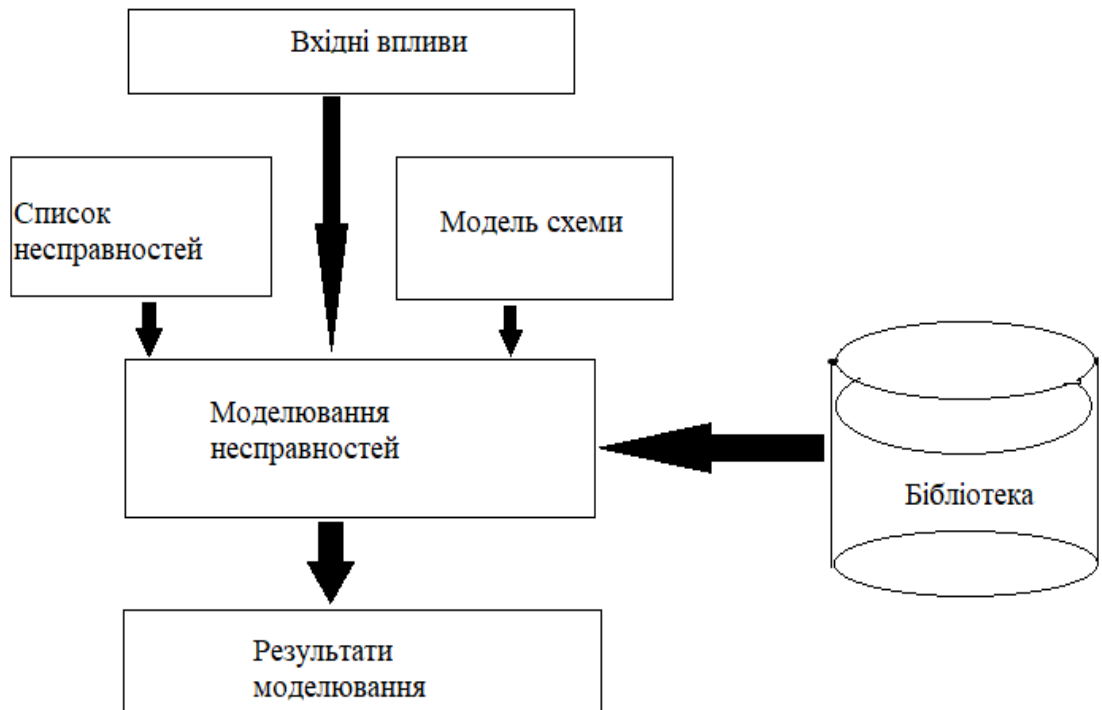


Рисунок 2.5 – Моделювання несправностей

Тут несправності із скороченого списку вводяться в модель схеми, далі моделюється поведінка схеми на вхідному наборі з подальшим аналізом. Якщо несправність перевіряється (дає вихідну реакцію, що відрізняється від реакції справної схеми), то вона видаляється із списку та розглядається наступна. У зворотному випадку подається наступний вхідний набір. Процес продовжується до тих пір, поки список несправностей не стане пустим. Аналіз несправних схем може суттєво перевищувати час моделювання справних схем. Тому при моделюванні несправних схем вирішальним фактором є *швидкодія* (в той час як для справних схем важлива перш за все *адекватність*) [11].

Порівняно з моделюванням справних ДП при моделюванні несправностей потрібно виконати наступні дії:

- 1) Визначення безлічі несправностей, що моделюються;
- 2) Внесення впливу несправностей;
- 3) Розповсюдження впливу несправностей;
- 4) Визначення перевіряємості несправностей.

Відрізняють умовно перевіряємі несправності та перевіряємі несправності. Несправність вважається умовно перевіряємою, якщо в результаті моделювання на зовнішньому виході в несправній схемі встановлюється невизначене значення u , а в у справній – визначене значення 0 або 1. Несправність вважається перевіряємою, якщо в

результаті моделювання хоча б на одному зовнішньому виході справної та несправної схем встановлюються відміні визначені значення. Методи моделювання ДП з несправностями, в основному, відрізняються різною технікою розповсюдження несправностей.

2.8 Типові моделі несправностей

Поодинокі константні несправності (stack-at fault 0, stack-at fault 1). Один вхід логічного схеми приймає постійне значення 0 або 1.

Кратні константні несправності (multiple stack-at faults). Кілька входів логічної схеми приймають постійні значення сигналів.

Місткові несправності (bridge faults). Декілька ліній схеми, значення сигналів на яких не залежать один від одного в справній схемі, стають залежними в несправній.

Затримки (delay faults). Затримка поширення сигналу по одній або декількох лініях схеми.

Перемежовувачі несправності (transient faults). Викликаються змінами внутрішніх параметрів схеми. помилки виникають при деяких (не всіх!) станах схеми. Як правило, в результаті несправність проявляється як постійна.

2.9 Прояв константних несправностей на функціональному рівні

Константа 0 на вході кон'юнктора. Ця несправність призводить до того, що на вихід кон'юнктор дорівнює нулю при будь-якому наборі значень входів. Якщо кон'юнктор відноситься до підсхеми кон'юнктивні чинників, то його вихід здійснюється з входами кон'юнктор підсхеми покриттів кон'юнкція, і в підсумку один або кілька виходів цієї підсхеми приймають значення 0. Якщо кон'юнктор відноситься до підсхеми покриттів кон'юнкція, то його вихід ототожнюється з одним або декількома входами рівня диз'юнкцій. Значить, дана несправність призводить до кратної несправності «константа 0» на рівні диз'юнкцій. На функціональному рівні це призводить до зникнення однієї або декількох кон'юнкція з однієї або декількох ДНФ.

Константа 1 на вході кон'юнктора. На функціональному рівні ця несправність призводить до того, що з кон'юнкції, яка реалізується даним кон'юнктор, зникає буква (якщо кон'юнктор відноситься до підсхеми кон'юнктивні факторів) або кілька букв (якщо він відноситься до підсхеми покриттів кон'юнкція). В результаті зникає одна або кілька букв з однієї або декількох кон'юнкція системи ДНФ.

Константа 0 на вході диз'юнктора. На функціональному рівні ця несправність призводить до зникнення кон'юнкції з однієї або декількох ДНФ.

Константа 1 на вході диз'юнктора. Ця несправність призводить до того, що на виході одного або декількох диз'юнкторів стають рівними одиниці. На функціональному рівні це означає, що замість деяких функцій реалізується константа 1. Також цю несправність можна розглядати як зникнення всіх букв із деяких кон'юнкцій системи.

Кратна константна несправність на входах одного кон'юнктора. Якщо хоча б на одному вході є несправність «константа 0», то кратна несправність еквівалентна одиночній типу «Константа 0», тобто призводить до зникнення однієї або декількох кон'юнкцій. Якщо все несправності є несправностями типу «константа 1», це призводить до зникнення кількох букв в одній або декількох кон'юнкціях.

Кратна константна несправність на входах різних кон'юнкторів. Призводить до зникнення деяких кон'юнкцій, або до зникнення деяких букв з кон'юнкцій, або до поєднання того і іншого.

Кратна константна несправність на входах одного або декількох диз'юнкторів. Якщо хоча б на одному вході є несправність «константа 1», це призведе до зникнення всіх букв в одній або декількох кон'юнкцій. Якщо все несправності мають тип «константа 0», це призведе до зникнення кількох кон'юнкцій. У загальному випадку кратна константна несправність такого типу призводить до комбінації зникнення всіх букв в одній або декількох кон'юнкцій і зникнення однієї або декількох кон'юнкцій.

Кратна константна несправність на входах кон'юнкторів та диз'юнкторів. В загальному випадку призводить до комбінації зникнення всіх букв в одній або декількох кон'юнкцій і зникнення однієї або декількох кон'юнкцій

Отже, будь-яка одиночна або кратна константна несправність схеми, синтезованої за безизбиточною системою ДНФ факторизаційним методом, призводить або до зникнення кон'юнкцій, або до зникнення букв з кон'юнкцій, або до комбінації цих помилок.

2.10 Останні дослідження

За способом обліку затримок в моделях елементів процедури моделювання діляться на синхронні (без урахування затримок) і асинхронні (враховують затримки в моделях елементів).

За способом обліку перехідних процесів процедури моделювання діляться на виконавчі і багатозначні. Багатозначні методи моделювання дозволяють у процедурах синхронного моделювання вплив перехідних процесів в схемах на результат моделювання.

Серед процедур багатозначного моделювання особливо виділяють потрібне моделювання в алфавіті $\{0,1,X\}$, яке, зберігаючи достатню швидкодію, дозволяє проводити аналіз перехідних процесів в цифрових схемах.

У діагностичних завданнях переважно застосовуються процедури зворотної імплікації і аналогічні їм по обчислювальній складності. З [14] відомо, що обчислювальна складність процедур зворотної імплікації в середньому в 50-100 разів вище, ніж для задач прямої імплікації. У зв'язку з цим в діагностичних завданнях застосовуються переважно процедури синхронного моделювання (синхронні моделі, де час затримки дорівнює 0). Але, разом з тим, певний інтерес представляє розгляд процедур асинхронного інтерпретативного моделювання.

2.11 Мікроелектронні функціональні цифрові вузли комбінаційного типу

Інтегральні логічні елементи є основою для побудови цифрових пристроїв, що виконують більш складні операції і відносяться до класу комбінаційних пристроїв.

Основні з них: дешифратори і шифратори; мультиплекси і демультіплексори; двійкові суматори; цифрові компаратори та мажоритарні елементи; перетворювачі кодів та інші.

2.11.1 Суматори

Суматори – це цифрові функціональні пристрої, що призначені для виконання операції складання чисел, що представлені у різних кодах.

За характером дії суматори діляться на комбінаційні, що не мають елементів пам'яті, та накопичуючі – запам'ятовуючі результати обчислень при знятті вхідних сигналів.

Надалі будуть розглядатися тільки комбінаційні суматори, на основі яких виконується більшість сумуючих (підсумкових) ІС.

Суматор за модулем два – це пристрій з двома входами (a і b), на виході якого сигнал «1» з'являється тільки в тому випадку, коли на входах діють протилежні сигнали, тобто «0» і «1». Суматор не володіє пам'яттю. Його логічне рівняння має вигляд:

$$y = \bar{a}b \cup a\bar{b} \quad (2.8)$$

Таблиця 2.7 - Таблиця істинності суматора за модулем два

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

Назву «за модулем два» цей суматор отримав тому, що у відповідає значенню молодшого розряду при складанні одно розрядних двійкових чисел А і В. Побудуємо в базисі І-НЕ схему суматора за модулем два (рис. 2.6).

$$y = \bar{a}b \cup a\bar{b} = \overline{\overline{\bar{a}b} \cup \overline{a\bar{b}}} = \overline{\overline{\bar{a}b} \cdot \overline{a\bar{b}}} = \overline{\bar{a}b \cdot a\bar{b}} \quad (2.9)$$

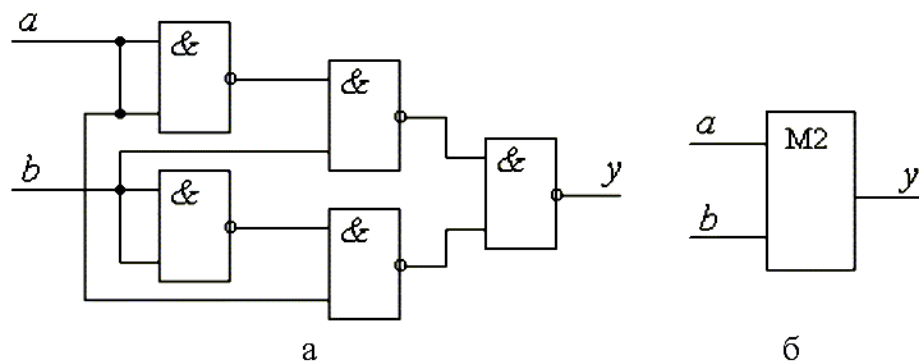


Рисунок 2.6 – Реалізація суматора за модулем два: а) принципальна схема;
б) функціональна схем

Напівсуматор забезпечує операцію складання двох одно розрядних двійкових чисел а і b. Так як при a=1 та b=1 отримуємо перенос одиниці в наступний розряд, напівсуматор повинен мати два виходи: з одного знімається сигнал суми за модулем два, а з іншого – сигнал переносу.

Таблиця істинності напівсуматора та його логічні рівняння мають вигляд:

$$S' = \bar{a}b \cup a\bar{b} \quad (2.10)$$

$$P' = ab \quad (2.11)$$

Таблиця 2.8 – Таблиця істинності напівсуматора

a	b	S'	P'
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Реалізацію напівсуматора в базисі І-НЕ представлено на рисунку 2.7.

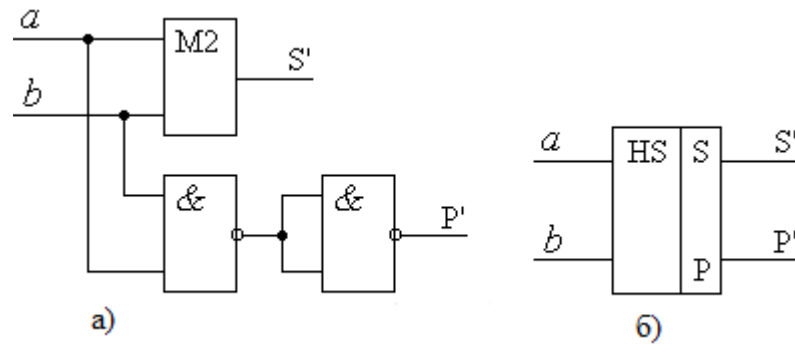


Рисунок 2.7 – Схема напівсуматора: а) принципальна схема; б) функціональна схема

Повний суматор – це пристрій для складання трьох одно розрядних двійкових чисел a , b , c , де c – сигнал переносу із попереднього молодшого розряду. Має два виходи: S (сума) та P (перенос).

Повний суматор можна побудувати з двох напівсуматорів (рис. 2.8), звідси і назва – напівсуматор, використовуючи наступні логічні рівняння:

$$S_i = \bar{P}_i \bar{a}_i b_i \cup \bar{P}_i a_i \bar{b}_i \cup P_i \bar{a}_i \bar{b}_i \cup P_i a_i b_i \quad (2.12)$$

$$P_{i+1} = \bar{P}_i a_i b_i \cup P_i \bar{a}_i b_i \cup P_i a_i \bar{b}_i \cup P_i a_i b_i = a_i b_i \cup P_i a_i \cup P_i b_i \quad (2.13)$$

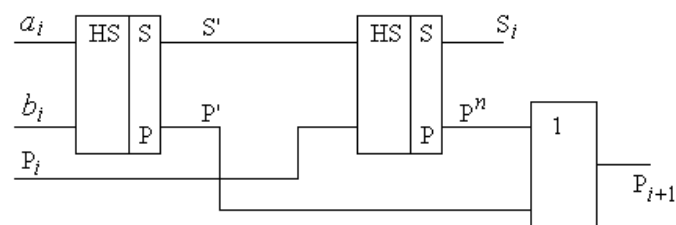


Рисунок 2.8 – Повний суматор

2.11.2 Дешифратори, шифратори, перетворювачі кодів

Дешифратор – комбінаційний пристрій, що дозволяє перетворювати n-розрядний двійковий код в позиційний 2^n -розрядний код. Має n входів та 2^n або менше виходів. В залежності від вхідного набору сигнал 1 з'явиться тільки на одному певному виході, а на всіх інших виходах будуть сигнали 0.

Логічні функції виходів дешифратора:

$$y_0 = \overline{x_3 x_2 x_1} \quad (2.14)$$

$$y_1 = \overline{x_3 x_2} x_1 \quad (2.15)$$

$$y_2 = \overline{x_3} x_2 x_1 \quad (2.16)$$

$$y_3 = \overline{x_3} \overline{x_2} x_1 \quad (2.17)$$

$$y_4 = \overline{x_3} x_2 \overline{x_1} \quad (2.18)$$

$$y_5 = \overline{x_3} \overline{x_2} \overline{x_1} \quad (2.19)$$

$$y_6 = \overline{x_3} x_2 x_1 \quad (2.20)$$

$$y_7 = \overline{x_3} \overline{x_2} \overline{x_1} \quad (2.21)$$

Таблиця істинності повного дешифратора на три входи представлена в таблиці нижче.

Таблиця 2.9 – таблиця істинності повного дешифратора на три входи

Номер вхідного набору	Входи			Виходи							
	x_3	x_2	x_1	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

За способом реалізації дешифратори можуть бути лінійними, прямокутними та пірамідальними. Більш досконаліми є пірамідальні дешифратори, що відносяться до багатоступеневих структур та містять ряд логічних елементів для виділення загальних частин функцій.

Нижче в таблиці 2.10 дана порівняльна оцінка лінійних, пірамідальних та прямокутних дешифраторів за апаратними затратами $N_{ле}$ у перерахунку на двоходові логічні елементи (ЛЕ) для m-розрядного коду.

Таблиця 2.10

Тип дешифратора		m						
		2	3	4	5	6	7	8
N _{ле}	Лінійний	4	16	48	128	320	768	1792
	Пірамідальний	4	12	28	60	124	252	508
	Прямокутний	4	12	24	48	88	164	304

Як видно з таблиці 2.10, перевага багатоступеневих дешифраторів помітно зростає із збільшенням m . В спеціалізованих ІС тим не менш перевагу часто віддають більш простим лінійним (одноступеневим) дешифраторам, які, до того ж, мають підвищену швидкодію.

Шифратор – комбінаційний пристрій, перетворюючий керуючий сигнал на одному із входів у відповідний двійковий код.

Для шифратора на чотири входи та два входи логічні рівняння в ДНФ, отримані із таблиці, будуть наступними:

$$y_1 = \overline{x_1}x_2\overline{x_3}x_4 \cup x_1\overline{x_2}x_3\overline{x_4}; \quad (2.22)$$

$$y_2 = \overline{x_1}\overline{x_2}x_3\overline{x_4} \cup x_1\overline{x_2}x_3x_4. \quad (2.23)$$

Таблиця 2.11 – Таблиця істинності

Номер набору	x ₁	x ₂	x ₃	x ₄	y ₁	y ₂
0	0	0	0	1	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	0
3	1	0	0	0	1	1

Найбільше застосування шифратори знаходять в цифрових пристроях вводу інформації з пультів керування для перетворення десяткових чисел у двійковий код. При натисканні на клавішу на один із входів шифратора подається логічна одиниця (на інші – логічні нулі), на виході формується відповідний двійковий код.

Умовне позначення дешифратора та шифратора наведено на рисунку 2.9:

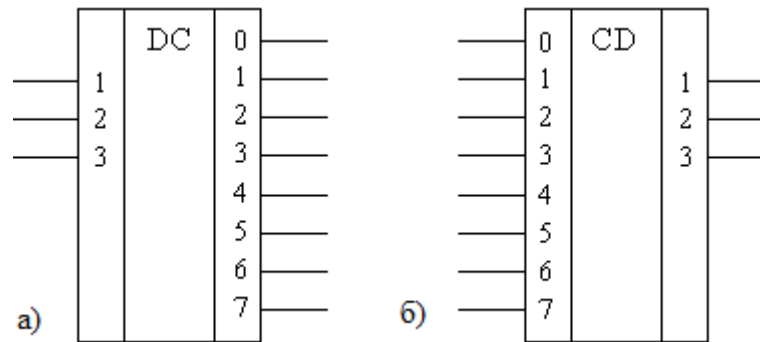


Рисунок 2.9 – Умовне позначення: а) дешифратор; б) шифратор

Перетворювачі кодів – це пристрої для автоматичного змінення за заданим алгоритмом відповідності між вхідним та вихідним кодами без змінення їхнього змістового вмісту.

Інакше, перетворювач коду – пристрій з m входами та n виходами, що взаємно та однозначно перетворює вхідні слова із деякого алфавіту $\{X_1, X_2, \dots, X_p\}$ та вихідні слова іншого алфавіту $\{Y_1, Y_2, \dots, Y_u\}$.

Задача перетворення кодів з'являється перед усім у зв'язку з необхідністю зведення цифрових пристроїв з різноманітними способами кодування в єдину систему.

Для перетворення паралельних двійкових кодів можна побудувати достатньо прості перетворювачі на комбінаційних логічних схемах. Однак на практиці це часто здійснюється алгоритмічним шляхом із використанням запам'ятовуючих пристроїв.

2.11.3 Мультиплексори, демюльтиплексори

Демюльтиплексор – (розподільник) пристрій, що передає сигнал, який поступив на його вхід x , на один із S виходів в залежності від керуючого сигналу, що заданий двійковим кодом.

Структура демюльтиплексора має вид (рис. 2.10):

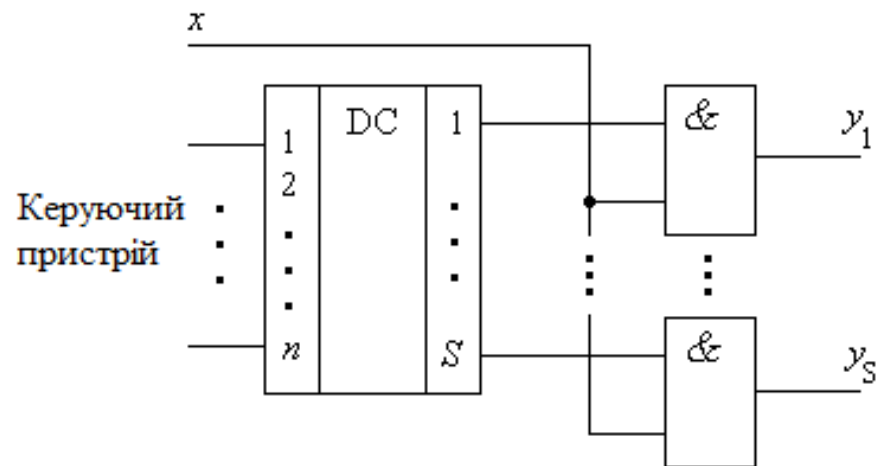


Рисунок 2.10 – Структура демультиплексора

Демультиплексори за своєю логікою роботи близькі до дешифраторів. Якщо на вхід x подати логічну одиницю, то показаний на рис. 2.10 демультиплексор перетворюється на дешифратор. Тому деякі промислово випускаємі дешифратори можуть виконувати функції демультиплексорів

Мультиплексор – Пристрій для комутації інформації, що поступає по декільком вхідним каналам, на один вихідний канал в залежності від керуючого сигналу, що заданий двійковим кодом.

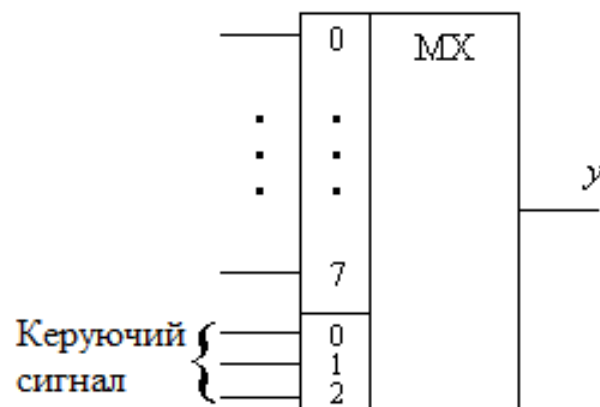


Рисунок 2.11 – Умовне позначення мультиплексора

Якщо мультиплексор має n -розрядний керуючий сигнал, то кількість комутуючих входів - 2^n (рис.2.11).

2.11.4 Цифрові компаратори

Компаратор – пристрій порівняння кодів чисел.

У загальному випадку компаратор паралельних кодів двох m -розрядних двійкових чисел представляє собою комбінаційну схему з $2m$ входами та трьома виходами («дорівнює», «більше», «менше»). При надходженні на входи кодів двох чисел, що порівнюються, сигнал логічної одиниці з'являється тільки на одному з виходів. У деяких випадках компаратор може мати менше трьох виходів.

Однорозрядний компаратор має два входи на які одночасно надходять одно розрядні двійкові числа x_1 та x_2 , і три виходи ($=$, $>$, $<$).

З таблиці істинності логічні рівняння компаратору при порівнянні x_1 з x_2 виходять у вигляді:

$$y^= = x_1 x_2 \cup \overline{x_1 x_2}, \quad (2.24)$$

$$y^> = x_1 \overline{x_2}, \quad (2.25)$$

$$y^< = \overline{x_1} x_2. \quad (2.26)$$

Таблиця 2.12 – Таблиця істинності

x_1	x_2	$y^=$	$y^>$	$y^<$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Реалізація такого компаратору в базисі І-НЕ призводить до схеми, що показана на рис. 2.12. Багаторозрядні компаратори зазвичай виконують на базі однорозрядних. При цьому використовується принцип послідовного порівняння розрядів багаторозрядних чисел, починаючи з їх старших розрядів, так як уже на цьому етапі, якщо $x_{1m} > x_{2m}$, задача може бути вирішена однозначно, і порівняння наступних за старшими розрядів не потребується.

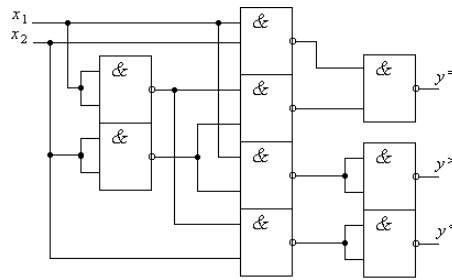


Рисунок 2.12 – Однорозрядний компаратор двійкових чисел

2.12 Висновки до розділу 2

У другому розділі проаналізовано основні задачі проектування систем на кристалі. Розглянуто основні логічні елементи для СНК, їх параметри, способи реалізації. Проаналізовано формули та логіку, за якими працюють всі елементи системи. Розглянуто типові несправності. Розглянуто мікроелектронні функціональні цифрові пристрої. Проведено аналіз задач моделювання з несправностями. Надано параметри цифрових інтегральних логічних схем. Проведено класифікацію польових транзисторів та розглянуто логічні схеми транзисторно-транзисторної логіки.

РОЗДІЛ 3

ВБУДОВАНЕ ТЕСТУВАННЯ МЕТОДОМ ГРАНИЧНОГО СКАНУВАННЯ JTAG

На сьогоднішній день найбільш затребуваним є саме реалізація методу граничного сканування за допомогою JTAG саме за свою багатофункціональність та великий спектр вирішуваних задач.

3.1 JTAG-інтерфейс та метод граничного сканування

Історично інтерфейс JTAG (Joint Test Action Group) з'явився як розвиток робіт європейської групи (JETAG) дослідників проблем розробки придатної для тестування апаратури в рамках спеціальної міжнародної групи, створеної з ініціативи фірми Texas Instrument, для вироблення стандарту на виробництво придатних для тестування ВІС. Результатом роботи цієї групи з'явився прийнятий в 1990 році стандарт IEEE Std. 1149.1 і його вдосконалена версія - стандарт IEEE Std.1149.1a (1993 рік). Запропонований стандарт мав два різних аспекти. Один полягав у розробці протоколу і принципів обміну інформацією між ВІС, з'єднаними в послідовний ланцюжок. Надалі для визначення цього аспекту стандарту будемо користуватися терміном «Транспортний механізм». А інший - в спеціальній (орієнтованій на тестування) організації зв'язку між основними схемами кристала ВІС і її зовнішніми контактами. Така організація зв'язку з цим дозволяє передавати значення сигналів на вихідних контактах в транспортний механізм ланцюжка і навпаки. Це дає можливість використовувати кордону ВІС для задач тестування їх міжз'єдань без фізичного доступу до кожного її висновку. Такий підхід отримав назву методу граничного сканування (Boundary Scan Testing, BST). Термін «граничне сканування» видається більш точно відповідним задачі сканування стану кордону між основними елементами ВІС і обладнанням, розташованим поза ВІС, ніж використовуваний іноді термін «периферійне сканування». Малюнок 1 показує структурну організацію ВІС, підтримуючої метод BST, і дозволяє зрозуміти основну концепцію граничного сканування.

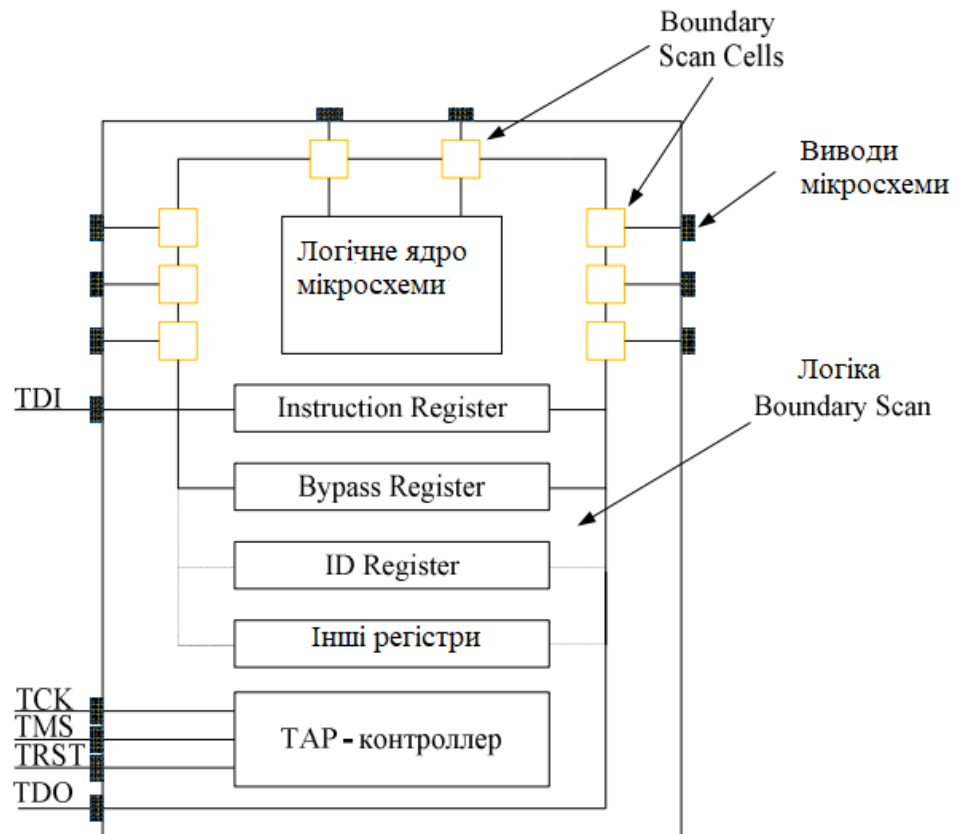


Рисунок 3.1 – Організація BIC, що використовує метод граничного сканування

Комірки сканування (Boundary Scan Cells, BSC) розміщені між кожним зовнішнім виводом мікросхеми та схемами кристала, утворюють власне BIC. Комірки, з одного боку, забезпечують прийом, збереження або видачу тестової інформації JTAG-ланцюжок, а з іншого - різні режими взаємодії між зовнішніми контактами BIC, що запам'ятовують тригерами осередку BSC і основними («логічне ядро») схемами кристала. Реалізація команд тестування базується на відповідності довжини регістрів прийому / передачі інформації (кількості BSC-комірок) і числа тестованих контактів BIC, а також на налаштування взаємодії розрядів цих регістрів з внутрішньою структурою BIC або станом зовнішніх контактів BIC.

Метод BST замислювався для виконання наступних тестових процедур:

- перевірки функціональної працездатності BIC за допомогою вбудованих в них тестових ланцюгів;
- перевірки якості з'єднань між контактами різних BIC, змонтованих на друкованій платі;
- зчитування або установки сигналів на вихідних контактах BIC в штатному режимі роботи BIC.

Виконання тестових процедур передбачає сумісну роботу трьох основних компонентів:

- джерела тестових команд та даних (тестового приладу), яким зазвичай є програма ПК. Цей же ПК тоді виступає і в якості аналізатора результатів тестування;
- механізму інформаційного зв'язку ВІС, що тестуються, та ПК, який тестує. Транспортний механізм інтерфейсу JTAG як раз і передбачає послідовне переміщення тестових команд та даних від вихідних ланцюгів ПК через ланцюг послідовно поєднаних ВІС до вхідного ланцюга ПК;
- схем керування JTAG-інтерфейсом, вбудованих в кожен ВІС, що тестується та забезпечує інтерпретацію BSC-комірками тестових команд ПК (реалізація власно методу граничного сканування). Якщо передбачається тестування працездатності внутрішніх схем окремих ВІС друкованої плати, то в архітектуру ВІС повинні бути додатково вбудовані апаратура та спеціальні тестові процедури самотестування BIST. Тоді запуск цієї процедури (автоматично при ввімкненні живлення або/та за подачею зовнішньої команди) дозволить судити про працездатність внутрішніх схем ВІС. Інформація про справність або несправність ВІС буде передаватися по лініях JTAG-інтерфейсу.

Правильне розуміння організації тестових процедур передбачає розглядання взаємодії всіх трьох компонентів. Тестовий прилад є не тільки джерелом всіх тестових процедур, але й пристроєм, що задає та синхронізує будь-які дії в ланцюгу і в приладах, що тестуються. Багато особливостей організації роботи тестового приладу та побудови схем керування JTAG-інтерфейсом окремих ВІС визначаються організацією транспортного механізму. Разом з тим, тестові експерименти, що реалізуються ПК, визначаються можливостями, закладеними в структуру окремих ВІС. Тому подальший розгляд від транспортного механізму до аналізу тестових можливостей окремих ВІС та далі до комплексної процедури тестування друкованої плати.

3.2 TAP-порт

Інтерфейс займає чотири виводи МК. За термінологією JTAG, дані виводи утворюють TAP-порт (Test Access Port). До виводів TAP-порту відносяться:

- TMS: вибір режимів тестування. Даний вивід призначений для навігації по цифровому автоматі TAP-контролера.
- TCK: тестова синхронізація. Через цей вивід здійснюється синхронізація інтерфейсу JTAG.

- TDI: ввід тестових даних. Вхід зсувного регістру, в якості якого може виступати регістр інструкції або регістр даних (ланцюги сканування).
- TDO: вивід тестових даних. Вихід зсувного регістру, тобто регістру інструкції або регістру даних.

Стандартом IEEE 1149.1-2001 також оговорений опціональний сигнал скидання TRST.

Якщо Fuse-біт JTAGEN знаходиться в незапрограмованому стані або рівний одиниці біт відключення JTAG, інтерфейс JTAG буде знаходитись у вимкненому стані. В такому випадку, виводи TAP-порту працюють як звичайні лінії вводу-виводу, а TAP-контролер знаходиться в стані скидання. Після дозволення роботи, на входах TAP-порту активізується підтяжка до плюсу живлення, а інтерфейс JTAG становиться готовим до операцій граничного сканування.

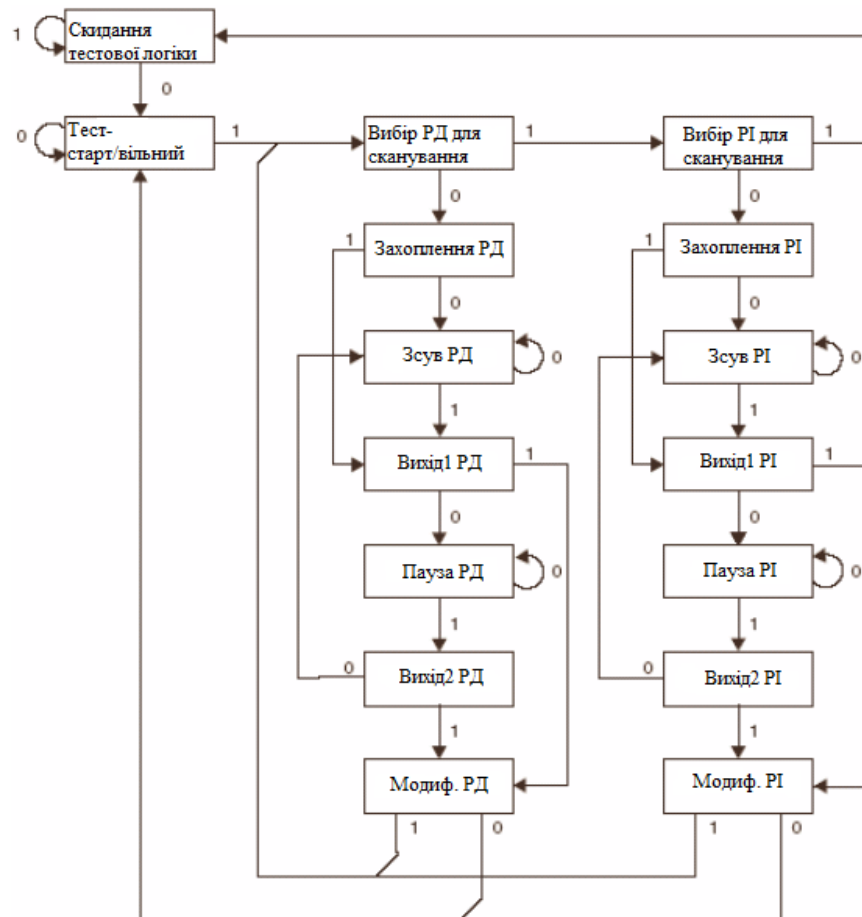


Рисунок 3.2 – Автомат станів TAP-контролера; РД – операції над даними; РІ – операції над інструкціями

TAP-контролер є кінцевим автоматом (16 станів), який керує роботою схеми граничного сканування. Зміна станів (см. рис. 3.2) відбувається в залежності від стану входу

TMS під час зростаючого фронту TСК. Початковим станом після скидання при подачі живлення є «Скидання тестової логіки».

Якщо припустити, що поточним станом є «Тест-старт/вільний», то можливі наступні сценарії використання інтерфейсу JTAG:

Через вхід TMS за зростаючими фронтами TСК вводиться послідовність 1, 1, 0, 0, яка викликає перехід в стан «Зсув РІ». В цьому стані за зростаючими фронтами TСК вводяться чотири біта в регістр JTAG-інструкції через вхід TDI. По ходу вводу 3 молодших бітів, для утримання стану «Зсув РІ», вхід TMS необхідно утримувати в низькому стані. Ввід старшого біта інструкції виконується при виході із стану «Зсув РІ» установкою високого рівню на вході TMS. По мірі вводу інструкції через вхід TDI, на виході виводиться код стану «Захоплення РІ»: 0x01. JTAG-інструкція обирає відповідний регістр даних в якості шляху між виводами TDI та TDO, а також керує схемою, що оточує обраний регістр даних.

1. Через вхід TMS передається послідовність 1, 1, 0, яка повторно вводить стан «Тест-старт/вільний». Інструкція фіксується на паралельних виходах шляху зсувного регістру в стані «Модиф. РІ». Для навігації по автоматі станів можуть використовуватися тільки стани «Вихід1 РІ», «Пауза РІ» та «Вихід2 РІ».
2. Через вхід TMS за зростаючими фронтами TСК вводиться послідовність 1, 0, 0, яка визиває перехід у стан «Зсув РД». Знаходячись в цьому стані, за зростаючими фронтами TСК через вхід TDI виконується модифікація обраного регістру даних. Для утримання стану «Зсув РД» вхід TMS необхідно утримувати в низькому стані при вводі всіх біт, окрім старшого. Старший біт даних вводиться при виході із стану, ініційованого установкою високого рівню на вході TMS. В процесі вводу даних в регістр даних через вхід TDI, стан паралельних входів, захоплений у стані «Захоплення РД» виводиться на виході TDO.
3. Через вхід TMS передається послідовність 1, 1, 0, яка повторно вводить стан «Тест-старт/вільний». Якщо обраний регістр даних оснащений паралельними виходами з фіксацією, то фіксація виконується у стані «Модиф. РД». Для навігації по автоматі станів можуть використовуватися тільки стани «Вихід1 РД», «Пауза РД» та «Вихід2 РД».

Як показано на автоматі станів, стан «Тест-старт/вільний» не треба вводити між вибором JTAG-інструкції та використанням регістрів даних.

3.3 Регістри даних

Між виводами TDI та TD можуть бути включені наступні регістри даних:

1. обхідний реєстр (реєстр А на рис. 3.3);
2. реєстр ідентифікації МК (реєстр С на рис. 3.3);
3. ланцюг граничного сканування (реєстр D на рис. 3.3);
4. реєстр даних PDI (реєстр В на рис. 3.3).

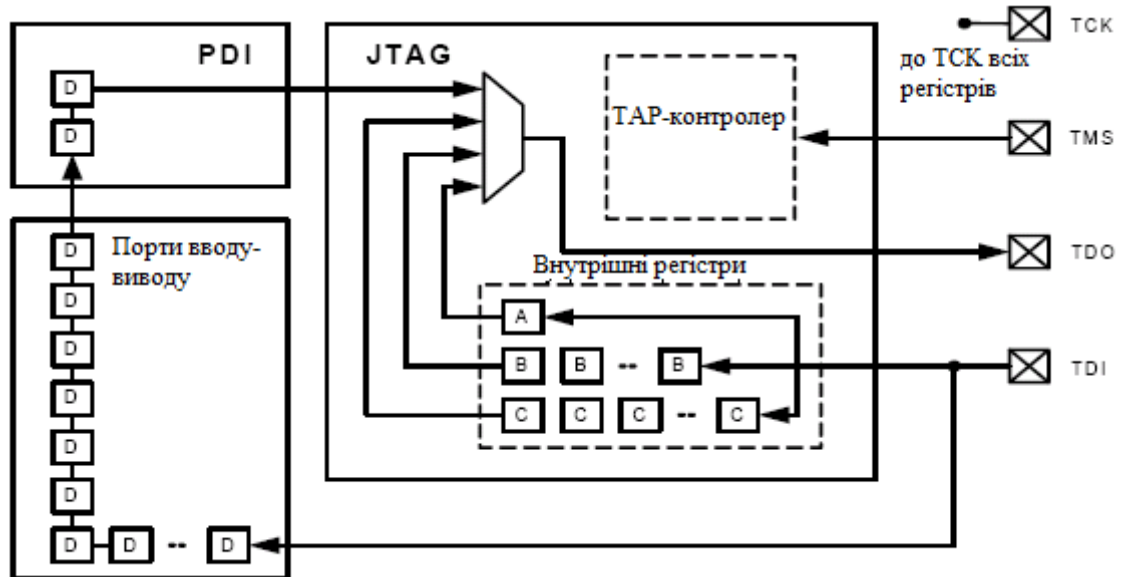


Рисунок 3.3– Огляд JTAG реєстра даних

3.3.1 Обхідний реєстр

Обхідний реєстр утворений одно каскадним зсувним реєстром. Коли обхідний реєстр обирається в якості шляху між виводами TDI та TDO, при виході контролера із стану «Захоплення РД», реєстр приймає нульове значення. Обхідний реєстр може використовуватися для мінімізації довжини ланцюга сканування під час тестування іншої мікросхеми.

3.3.2 Реєстр ідентифікації МК

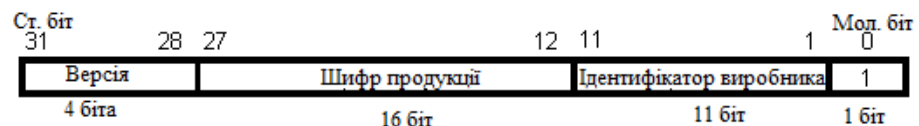


Рисунок 3.4 – Реєстр ідентифікації МК

Версія – 4-бітне значення, що дозволяє ідентифікувати версію компонента. Тут використовується JTAG нумерація версій. Версії А відповідає код 0x0, версії В – 0x1 і т.д.

Шифр продукції – це 16-бітний код, що дозволяє встановити назву мікросхеми. Узнати числове значення шифру продукції мікроконтролера, що зацікавив, можна в його документації.

Ідентифікатор виробника – це 11-бітний код, що дозволяє встановити виробника мікросхеми.

3.3.3 Регістр даних PDICOM

Регістр даних PDICOM – 9-бітний регістр, який використовується для послідовно-паралельного та паралельно-послідовного перетворення даних між JTAG TAP та PDI.

3.4 Механізм граничного сканування

Механізм граничного сканування визначається організацією скануючих BSC-комірок. Вони забезпечують реалізацію перелічених нижче режимів.

1. Режими самотестування ВІС, всередину якої передається інформація з BSC-комірок. Інформація відповідає поступившим з JTAG-ланцюга командам або даним. Подальша послідовність дій визначається поступившою командою та може відповідати фіксації в комітках BSC результуючої інформації, яку також можна передати до JTAG-ланцюга.

2. Режими тестування з'єднання ВІС між собою. В цьому режимі, так само як і в попередньому випадку, зовнішні контакти відключаються від внутрішніх схем ВІС. Однак, на відміну від попереднього режиму, інформація із комірок BSC надходить не всередину ВІС, а назовні на її зовнішні вихідні контакти. Фіксація в осередках BSC, що надійшли на вхідні (або двоспрямовані) контакти, дозволяє судити про наявність або відсутність реального з'єднання відповідної групи контактів. Аналіз цієї інформації здійснюється тестуючим пристроєм шляхом її передачі по JTAG-ланцюгу.

3. Режим тестування штатної роботи ВІС. В цьому режимі зберігається потрібне з'єднання зовнішніх контактів ВІС та внутрішніх схем кристалу. Фіксація в осередках BSC значень сигналів від усіх контактів ВІС (у заданий з JTAG-ланцюга момент часу) та наступна їх передача в тестуючій пристрій дозволяє проектувальнику отримати інформацію, що його інтересує. Значення внутрішніх сигналів тестуємої системи становляться відомими без організації фізичного доступу до контактів контрольованих ВІС.

Структурна схема скануючих комірок, що дозволяють забезпечити реалізацію описаних вище режимів, в своїй основі має вид, наведений на рис. 3.3. Схема містить два мультиплексора та два D-тригери. Один з них, T1, є тригером в зсувному регістрі даних JTAG-ланцюга: в стані вводу/виводу даних інформація зсувається за ланцюгом від попереднього до наступної BSC-комірки. В цих же тригерах (за іншого стану вхідного мультиплексора) може фіксуватися вхідна інформація BSC-комірок. Тригер T2 є тригером-засувкою, що буферизує дані основного зсувного тригера. Робота комірки залежить від режиму використання. В робочому режимі інформація зі входу комірки передається на вихід, поєднуючи вихідний контакт ВІС з внутрішніми ресурсами ВІС, але при цьому вхідна інформація комірки може використовуватися в тригері T1 зсувного регістра. В більшості тестових режимів вхід і вихід комірки роз'єднані. Вхідна інформація комірки може при цьому фіксуватися в тригері T1, а вихідна інформація буде визначатися вмістом тригера T2.

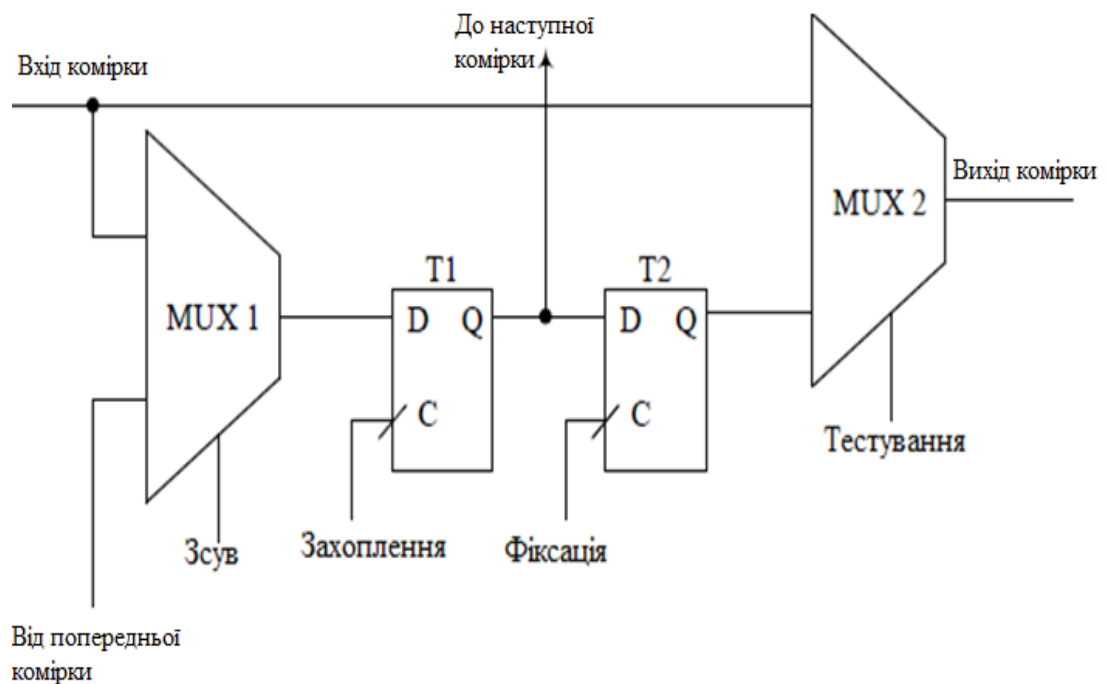


Рисунок 3.5 – Структура комірки сканування

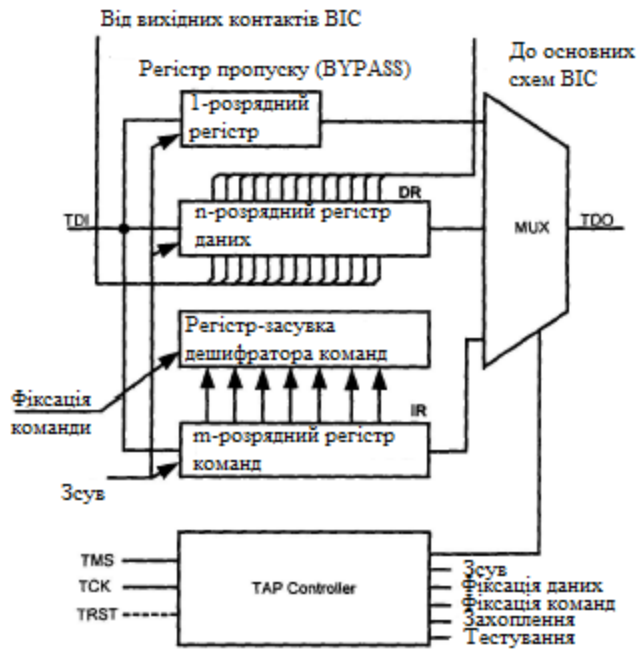


Рисунок 3.6 – Структура пристрою керування граничного сканування

3.5 Ланцюг граничного сканування

Ланцюг граничного сканування дозволяє керувати логічними рівнями на лініях вводу-виводу та контролювати їх. Щоб гарантувати передбачуваність поведінки мікроконтролера в процесі і після виконання інструкцій EXTEST, CLAMP та HIGHZ, мікроконтролера автоматично переходить до стану скидання. В цьому стані відключаються зовнішні генератори, аналогові модулі та програмно задані налаштування портів (в тому числі підтяжка до плюсу або мінусу, адаптивна підтяжка та виходи типу «Монтажне I/АБО»). Необхідно зауважити, що інструкції SAMPLE та PRELOAD не впливають на поточний стан мікроконтролера та його портів.

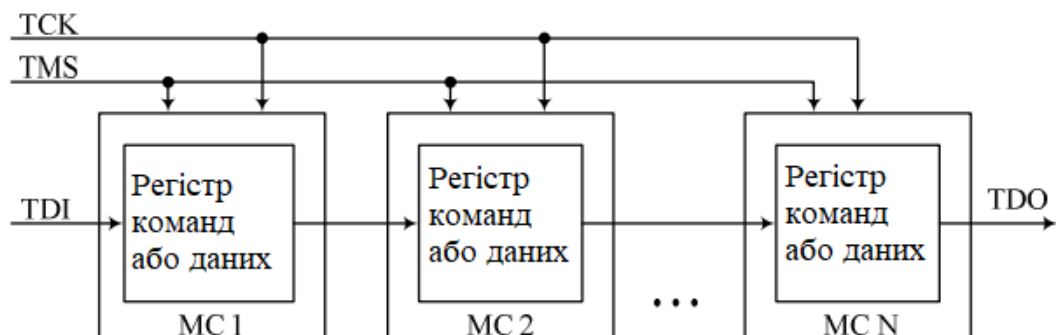


Рисунок 3.7 – Приклад об'єднання в JTAG-ланцюг

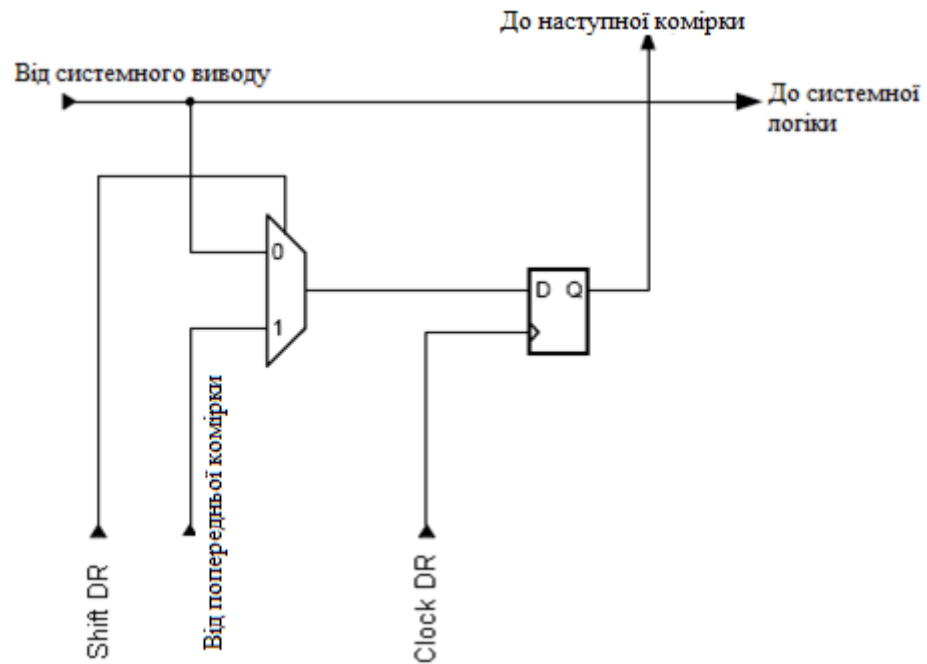


Рисунок 3.9 – Комірки з можливостями контролю входів

3.6 Трансивер JTAG

Трансивер JTAG відповідає за послідовну передачу по чотирьох лініях вводу-виводу (TMS, TCK, TDI, TDO). Він підтримує виявлення умови BREAK, виявлення помилок паритету та генерацію паритету.

3.6.1 Дозвіл роботи

Для дозволу роботи інтерфейсу JTAG необхідно запрограмувати Fuse-біт JTAGEN та скинути біт JTAG Disable в регістрі керування МК. Стандартно, fuse-біт JTAGEN запрограмований і робота інтерфейсу JTAG дозволена. Коли в JTAG регістр інструкції вводиться інструкція PDICOM, в якості регістра даних, включеного між виводами TDI і TDO, обирається комунікаційний регістр PDI. В цьому режимі, інтерфейс JTAG може використовуватися для доступу до можливостей програмування та налаштування інтерфейсу PDI

3.6.2 Відключення

Відключити інтерфейс JTAG можна або переведенням fuse-біта JTAGEN до незапрограмованого стану, або шляхом програмного встановлення біта JTAG Disable в регістрі керування мікроконтролером.

3.6.3 Набір інструкцій JTAG

Набір інструкцій JTAG мікроконтролерів XMEGA складається із восьми інструкцій, що відносяться до граничного сканування і доступу до контролера PDI в цілях програмування енергонезалежної пам'яті. Регістр інструкції є 4-бітним. Ввід та вивід молодшого біта здійснюється першим у всіх зсувних регістрів. Код операції для кожної інструкції показано відразу після її назви у шістнадцятковому форматі.

EXTEST; 0x1

Обов'язкова інструкція, призначена для вибору ланцюга граничного сканування (дозволяє протестувати зовнішню, підключену до МК AVR, схему) як регістр даних. Через ланцюг граничного сканування можна змінити напрямок (DIR) і стан (OUT) будь-якої лінії введення-виведення, а також проконтролювати налаштування напрямку лінії і її фактичний стан. Вміст виходів з фіксацією ланцюга граничного сканування стає доступним на виходах відразу після завантаження в регістр інструкції команди EXTEST.

Активні стани:

1. Захоплення РД: виконується вибірка даних на зовнішніх виводах для зчитування через ланцюг граничного сканування.
2. Зсув РД: дані в ланцюгу граничного сканування зсуваються під керуванням входу ТСК.
3. Модиф РД: дані із ланцюга сканування подаються на виводи виходів.

IDCODE; 0x3

Обов'язкова інструкція, призначена для вибору 32-бітного регістра ідентифікації МК як регістра даних.

Регістр ідентифікації містить наступні дані: номер версії, номер МК і JEDEC-код виробника. Дана інструкція є інструкцією за замовчуванням після подачі живлення.

Активні стани:

1. Захоплення РД: виконується вибірка вмісту регістра IDCODE в регістр ідентифікації МК.
2. Зсув РД: ланцюг сканування IDCODE передається зсувом під керуванням входу ТСК.

SAMPLE/PRELOAD; 0x2

Обов'язкова інструкція, призначена для попереднього завантаження виходів з фіксацією і опитування стану входів/виходів без порушення роботи системи. Однак необхідно враховувати, що виходи з фіксацією не пов'язані з выводами. В якості регістру даних виступає ланцюг граничного сканування.

Кожна з інструкцій SAMPLE і PRELOAD виконують одні ті ж дії, що пояснює причину загального коду операції у них і об'єднання в одну інструкцію.

Активні стани:

1. Захоплення РД: виконується вибірка даних на зовнішніх выводах для читання через ланцюг граничного сканування.
2. Зсув РД: передача зсувом ланцюга граничного сканування під керуванням входу ТСК.
3. Модиф РД: дані із ланцюга граничного сканування подаються на виходи з фіксацією. Треба враховувати, що виходи з фіксацією не пов'язані з выводами.

BYPASS; 0xf

Обов'язкова інструкція що призначена для вибору регістра обходу в якості регістра даних.

Активні стани:

1. Захоплення РД: завантажує логічний «0» в регістр обходу.
2. Зсув РД: регістр обходу включається між выводами TDI та TD.
3. Яз зсувний регістр.

CLAMP; 0x4

Опційна інструкція, яка дозволяє назначити стан выводів МК за станом попередньо завантажених фіксацій. В якості регістра даних обирається регістр обходу.

Активні стани:

1. Захоплення РД: завантажує логічний «0» в регістр обходу.
2. Зсув РД: регістр обходу працює як зсувний регістр між выводами TDI та TDO.

HIGHZ; 0x5

Опційна інструкція, що призначена для переведення всіх виходів в неактивний стан (наприклад, у високоімпедансний). В якості регістра даних обирається регістр обходу.

Активні стани:

1. Захоплення РД: завантажує логічний «0» в регістр обходу.
2. Зсув РД: регістр обходу включається між виводами TDI та TD.
3. Яз зсувний регістр.

PDICOM; 0x7

Специфічна AVR-інструкція, призначена для використання порту JTAG TAP в якості альтернативного каналу доступу до інтерфейсу програмування і налаштування PDI.

Активні стани:

1. Захоплення РД: фіксація паралельних даних із PDI в регістрі даних PDICOM.
2. Зсув РД: передача зсувом регістра даних PDICOM під керуванням входу TCK.
3. Модиф РД: копіювання команд або операндів в паралельному форматі із регістра даних PDICOM до інтерфейсу PDI.

3.6.4 Формат послілки та символів

Трансивер JTAG використовує фіксований формат послілки. Послідовна послілка визначена як один символ, що складається з восьми бітів даних та наступного за ними одного біту паритету.

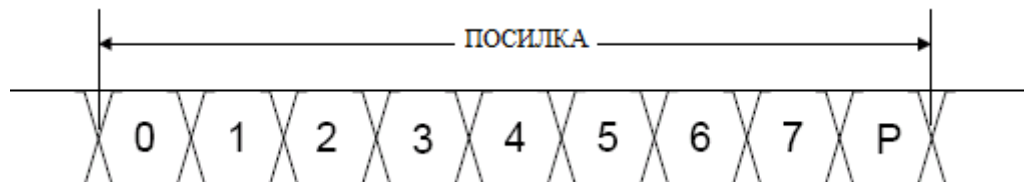


Рисунок 3.10 – Формат послідовної послілки інтерфейсу JTAG

Таблиця 3.1

(0-7)	Біти даних/команди (0...7), молодший біт передається першим
P	Біт паритету, використовується парний паритет

3.6.4.1 Спеціальні символи даних

Передбачено три спеціальні символи даних. У всіх цих символів біт паритету інвертований. Це необхідно для того, щоб при прийманні утворювалась помилка паритету. Символ BREAK (0xBB+P1) використовується зовнішнім програматором для того, щоб примусово перервати будь-яку операцію, що виконується контролером PDI, та перевести його в відомий стан. Символ DELAY (0xDB+P1) використовується контролером PDI для сигналізації програматору про те, що у нього немає готових даних програматора і що він не збирається виконувати передачу (тобто PDI знаходиться в режимі приймання).

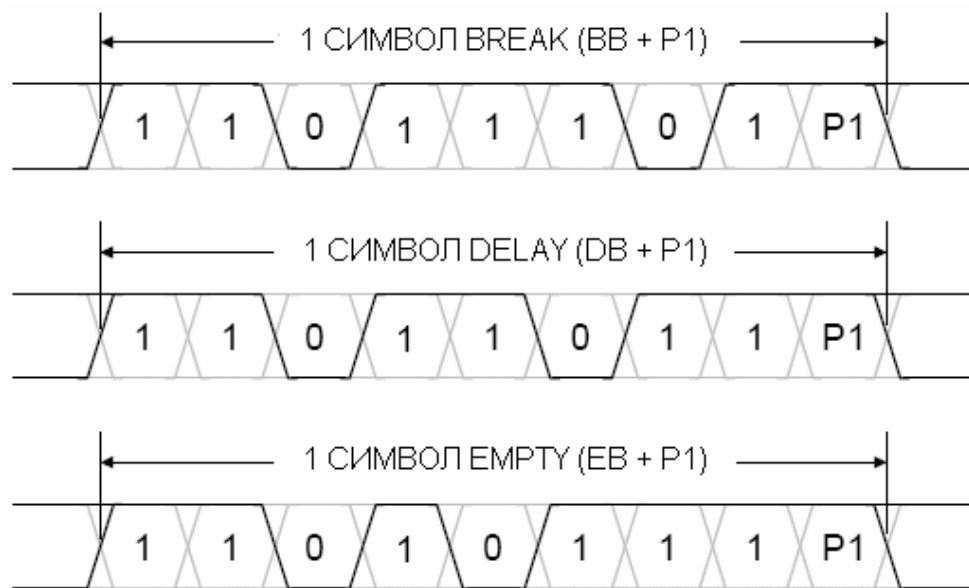


Рисунок 3.11 - Спеціальні символи даних

3.6.5 Послідовні передача та приймання

Інтерфейс JTAG використовує повнодуплексну передачу даних. Одночасно з введенням даних зсувом з лінії TDI, виконується вивід даних зсувом на лінію TDO. В свою чергу, інтерфейс TDI використовує напівдуплексну передачу даних. Таким чином, працюючи сумісно з контролером PDI, трансивер JTAG працює або в режимі передачі (TX), або в режимі приймання (RX).

Програматор і інтерфейс JTAG працюють синхронно з сигналом TCK, який генерується програматором. Залежність між фронтами синхронізації та вибіркою або зміною даних фіксована. На рис. 3.12 показано, що зміна рівнів на лініях TDI та TDO завжди відбувається по падаючому фронту TCK, а вибірка даних – по зростаючому фронту TCK.

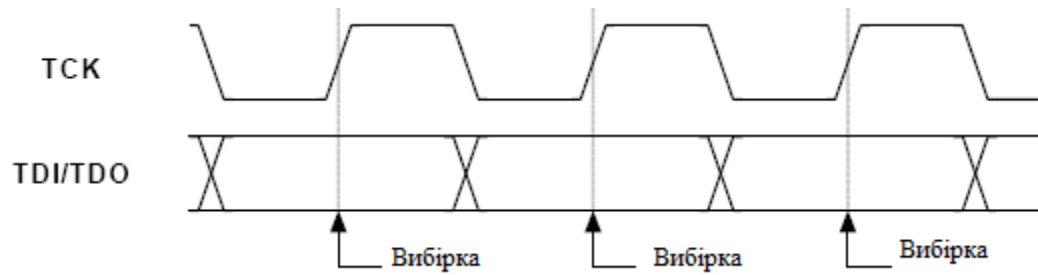


Рисунок 3.12 – Зміна та вибірка даних

3.6.6 Послідовна передача

Після ініціації передачі даних, в зсувний регістр в паралельному форматі поміщається байт даних, після чого він послідовно передається зсувом по лінії TDO. Сгенерований біт паритету передається відразу за бітами даних. Швидкість передачі залежить від сигналу TCK.

3.6.6.1 Сигналізація статусу

Якщо PDI перейшов в режим передачі (як відповідь на інструкцію LD) і під час введення TAP-контролером стану «Захоплення_РД» контролер PDI відправив запит на передачу, виконується завантаження дійсних даних в паралельному форматі в зсувний регістр і генерація біта паритету, який в подальшому буде переданий в стані «Зсув_РД» разом з байтом даних. Якщо PDI знаходиться в режимі прийому і TAP-контролер вводить стан «Захоплення_РД», в зсувний регістр поміщається байт EMPTY (0xEB) в паралельному форматі, а біт паритету встановлюється рівним одиниці (щоб викликати помилку паритету) і передається в подальшому разом з байтом даних в стані «Зсув_РД». Така ситуація виникає в ході прийому нормальної команди PDI і операнда.

Якщо PDI перейшов в режим передачі (як відповідь на інструкцію LD), але від контролера PDI запит на передачу ще не відправлений, TAP-контролер вводить стан «Захоплення_РД», байт DELAY (0xDB) завантажується в зсувний регістр, а біт паритету встановлюється рівним одиниці (щоб викликати помилку паритету) і передається разом з байтом даних в стані «Зсув_РД». Дана ситуація виникає в ході передачі даних, коли ще немає доступних даних для передачі.

На рисунку 3.13 показаний безперервний потік посилок даних з PDI (МК), які є результатом циклічного виконання інструкцій LD. В даному прикладі МК не може

повертати дані швидше одного байта за дві переданих посилки, тому що між посилками вставляються проміжні символи DELAY.

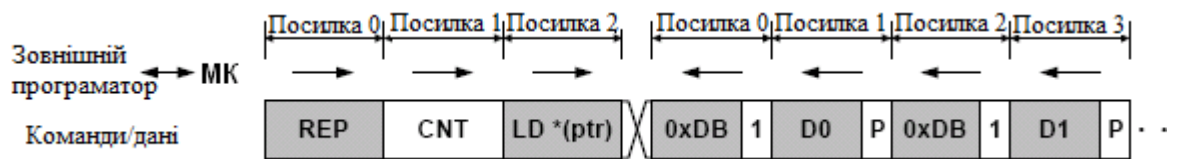


Рисунок 3.13 – Маркування готовності даних

Якщо посилка даних DELAY передається у відповідь на інструкцію LD, програматор повинен інтерпретувати це так, як ніби інтерфейс JTAG ще не мав даних для передачі під час попереднього стану «Захоплення_РД». Адекватною реакцією вибору програм в такому разі є ініціація повторюваних передач аж до прийому дійсного байта даних. Виконання інструкції LD передбачає повернення певної кількості дійсних посилок, а не просто кількості посилок. Таким чином, якщо програматор після передачі інструкції LD виявляє символ DELAY, інструкцію LD можна передавати повторно, тому що ще повинна бути відправлена відповідь на першу інструкцію LD.

3.6.7 Послідовний прийом

Під час прийому, приймач заповнює зсувний регістр з входу TDI вісьмома бітами даних і бітом паритету. При кожному прийомі дійсної посилки відбувається фіксація даних в паралельному форматі в стані «Модиф_РД».

3.6.7.1 Перевірка паритету

Блок перевірки паритету обчислює паритет (використовується парний режим) біт даних прийнятої посилки і порівнює його з прийнятим в послідовній посилці бітом паритету. Якщо буде виявлено помилку паритету, про це сигналізується контролер PDI.

3.6.7.2 Виявлення символу BREAK

Блок перевірки паритету активний як в режимі передачі, так і в режимі прийому. Якщо буде виявлено помилку паритету, прийнятий байт даних порівнюється з кодом

символу BREAK (при передачі якого штучно вводиться помилка паритету). Якщо буде виявлено символ BREAK, то про це сигналізується контролер PDI.

3.7 Структурна схема

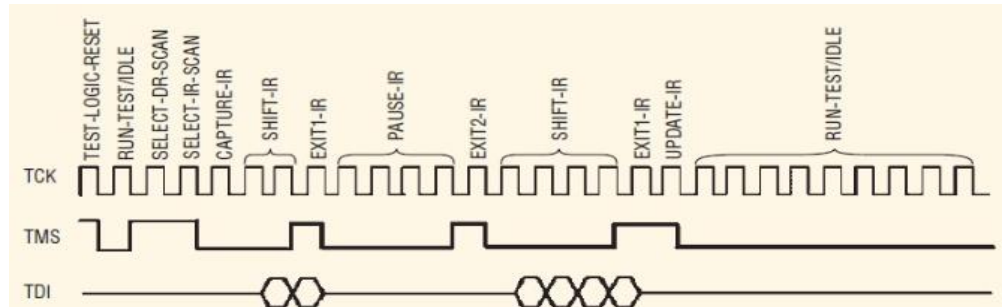


Рисунок 3.14 – Діаграма сигналів при виконанні переходів для завантаження команд

Якщо використовуючи спеціалізований САПР написати код роботи JTAG на Verilog, то результуюча діаграма буде такою, як показано на рисунку 3.15.

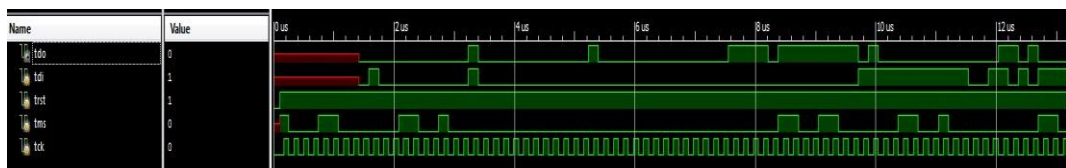


Рисунок 3.15 – Результуюча діаграма

Повна структурна схема пристрою JTAG зазначена нижче на рис. 3.16.

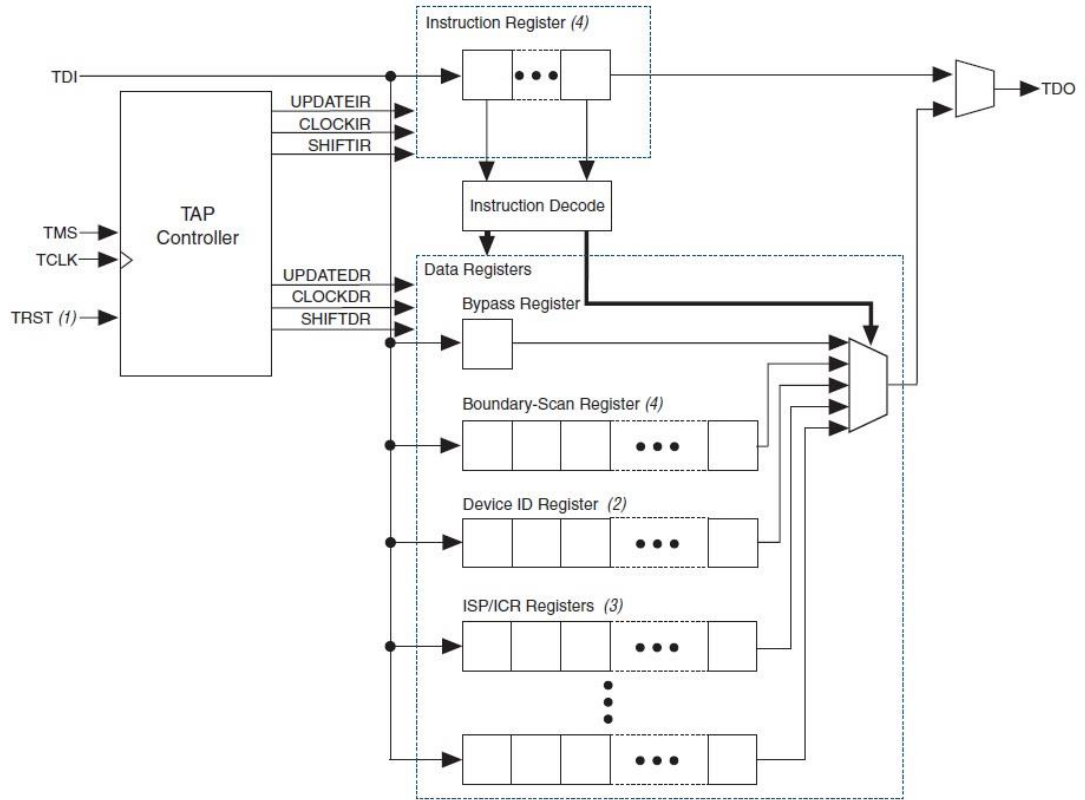


Рисунок 3.16 – Повна структурна схема пристрою JTAG

3.8 Місце JTAG-тестування в загальному процесі виробництва

JTAG-тестування виконується після того, як на виготовленій платі перевірено наявність вірних напруг живлення.



Рисунок 3.17 - Місце JTAG-тестування в загальному процесі виробництва

JTAG дозволяє виявляти непропай в BGA-корпусах, замикання, обриви, а також непрацюючі мікросхеми з цифровими інтерфейсами. Дуже важливо виявити всі дефекти, так як якщо неперевірена плата переходить на етап програмування, можуть виникнути проблеми з запуском пам'яті та периферії. При цьому буде важко встановити причину неполадок: невірні налаштування ПЗ або дефект монтажу. JTAG-тестування дозволяє попередити цю проблему.

3.9 Тестування аналогової частини плати

Взагалі-то прийнято вважати, що за допомоги JTAG аналогову частину неможливо протестувати. Стандарт IEEE 1149.1 розроблявся як засіб для тестування цифрової електроніки. Воно й зрозуміло, адже оперує він цифровими сигналами, тому при наявності на платі однієї або декількох ІС, що підтримують цей стандарт, можна протестувати зв'язки між цими компонентами, ланцюгами, що їх оточують (наприклад, на замикання), а крім того, зв'язку з ОЗП (SRAM, DRAM, SDRAM, DDR і ін.), ПЗП (послідовні, паралельні), логіку і інтерфейси. Однак на платі може бути присутня і аналогова частина, для якої периферійне сканування не призначене.

В останні кілька років багато проблем викликали високошвидкісні цифрові лінії зв'язку (наприклад, LVDS), які можуть містити аналогові складові, такі як ємності, що розв'язуються по змінному струму. Це значно обмежувало застосування стандарту IEEE 1149.1, який досить статичний. Однак рішення прийшло у вигляді стандарту IEEE 1149.6, що додає до звичної логіки периферійного сканування додаткові динамічні елементи. Зараз вже багато ІС на додаток до IEEE 1149.1 підтримують також і 1149.6, а системи автоматичного проектування тестів (наприклад, JTAG ProVision) успішно з такими пристроями працюють.

Крім того, цифро-аналогові ділянки плати при створенні додатків JTAG-тестування теж можуть тестуватися. Візьмемо, наприклад, АЦП - вихід цього пристрою часто йде на компонент з підтримкою периферійного сканування, отже, система тестування може зчитувати дані з АЦП в регістр периферійного сканування цього компонента. Таким чином, подаючи різні напруги на вхід АЦП (наприклад, граничні значення) можна створити тестовий додаток, що зчитує дані з цифрового виходу і перевіряє, чи укладаються вони в необхідний діапазон. А можна за допомогою того ж периферійного сканування управляти ЦАП, встановленим або на тестованій платі, або на оснащенні, і зчитувати дані з АЦП в рамках одного і того ж додатка. Звичайно, для такого роду додатків, можливо, буде потрібно створення додаткового оснащення, і автоматом їх не створити. Однак існують

засоби, які використовують для таких цілей вбудовану мову програмування, наприклад JTAG Functional Test (JFT).

3.10 Інфраструктура JTAG

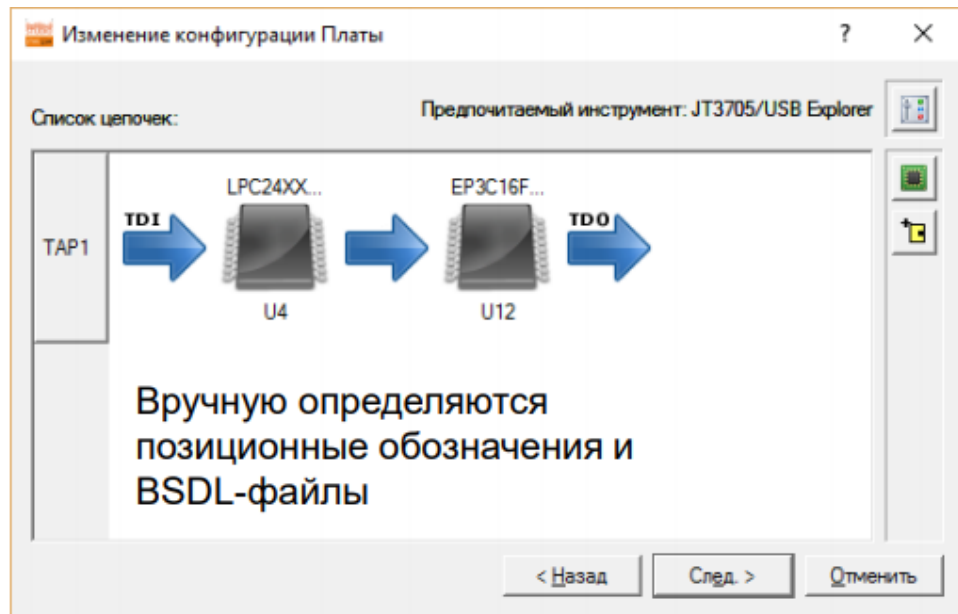


Рисунок 3.18 - Автовизначення JTAG-ланцюга

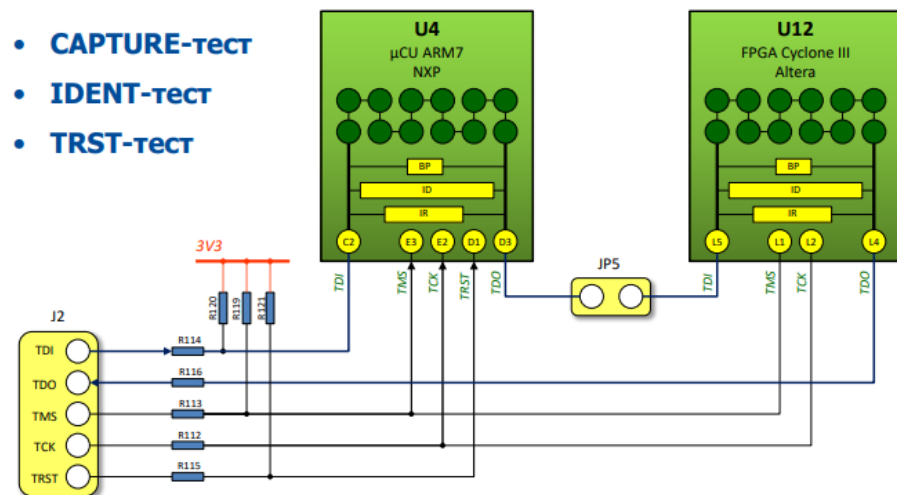


Рисунок 3.19 - Тестування інфраструктури

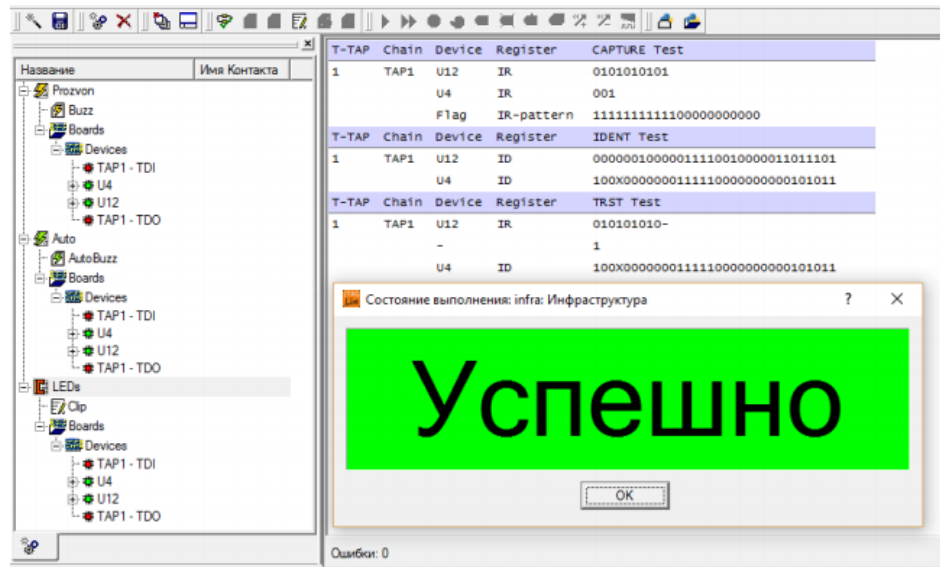


Рисунок 3.20 - Результат

3.11 CLIP: простий логічний аналізатор

Особливості:

- Виставляє та зчитує створені вручну вектори з пінів компонентів, що підтримують периферійне сканування.
- Вектори можна створювати в bin, hex, dec.
- Можна зберігати зчитану сигнатуру.
- Є графічне представлення тестових векторів в стилі логічного аналізатору.

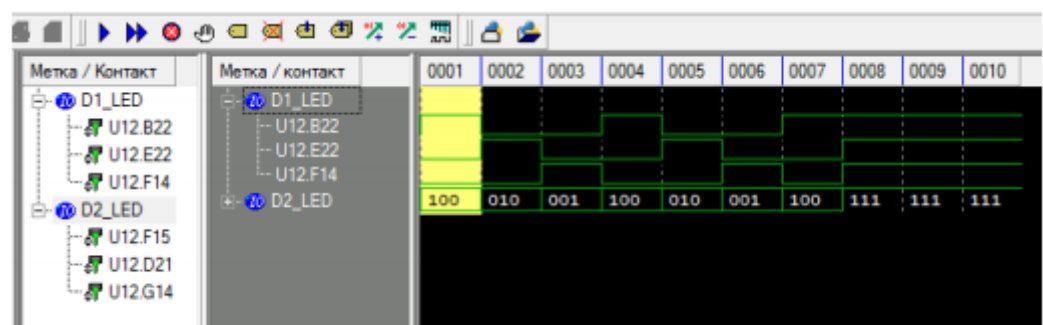


Рисунок 3.21 – Графічне представлення тестових векторів

3.12 Зчитування та перевірка зв'язків

З додатка В знаємо, як організовано «гірлянду» за допомоги CLIP та як саме розташовані лінії зв'язку з периферійними пристроями. На рисунках 3.22-3.23 показано послідовність дій при вивченні та порівнянні результатів при тестуванні контактів.

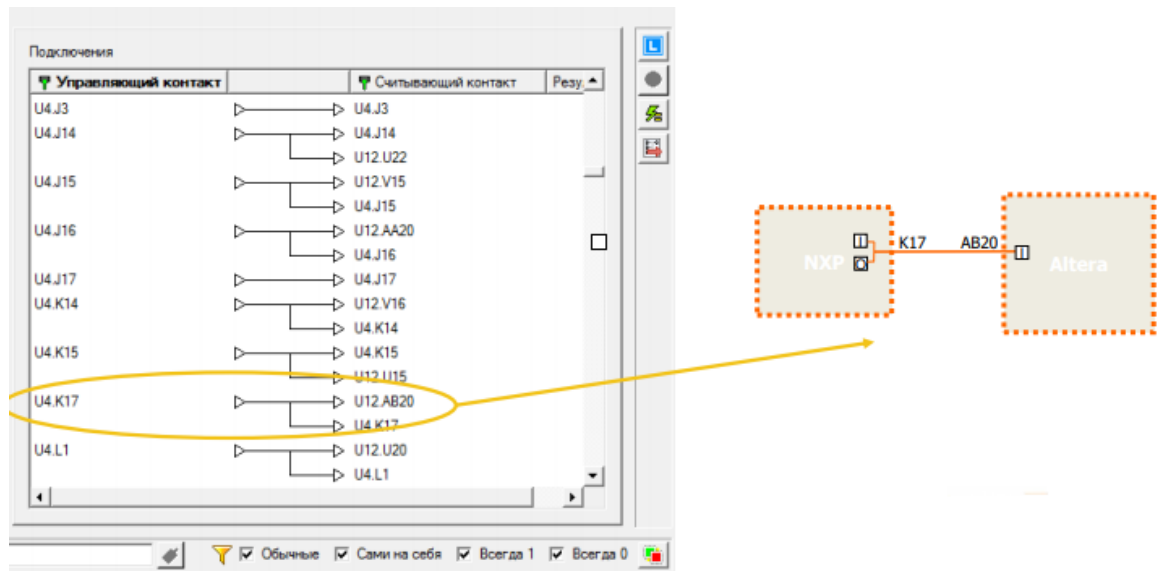


Рисунок 3.22 – Крок перший: вивчення

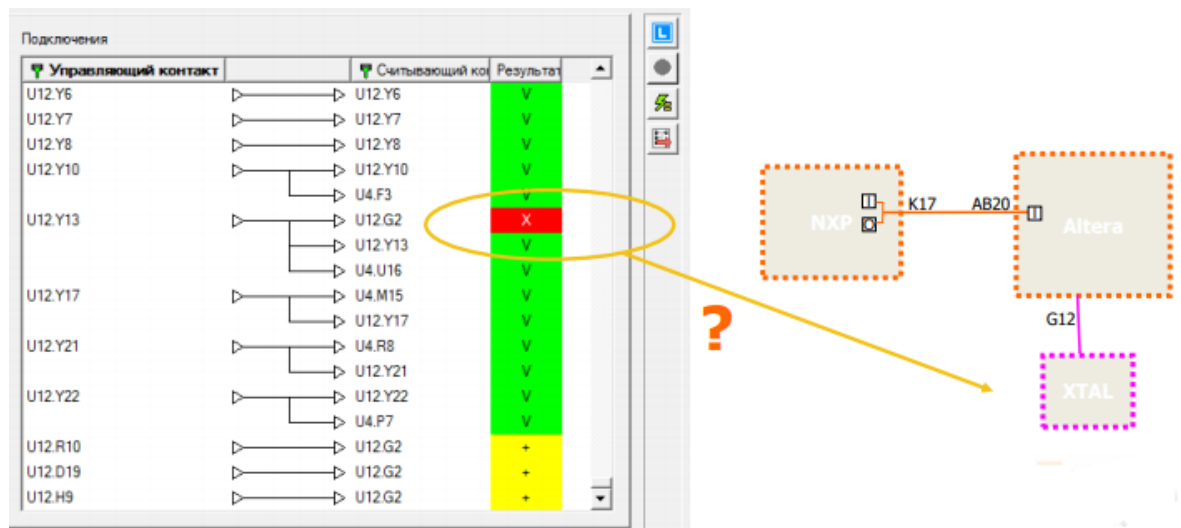


Рисунок 3.23 – Крок другий: повторне вивчення та порівняння з попередніми результатами

3.13 Мова JTAG Live Script

Особливості:

- Заснована на вбудованій мові програмування Python.
- Готові бібліотеки дозволяють працювати з виводами JTAG-компонентів плати та виконувати операції встановлення на них логічних даних, зчитування даних з ланцюгів, організувати шини.

Приклади готових функцій Script:

- **DriveLow(VarName)** – встановлює логічний нуль на вказаному піні. Приклад:
DriveLow("D201.B2(8)"), DriveLow("Decoder.G1").
- **DriveHigh(VarName)** – встановлює логічну одиницю на вказаному піні. Приклад:
DriveHigh("D201.B2(8)"), DriveHigh("Decoder.G1").
- **DeclareGroup(GrpName, GroupmemberList)** – дозволяє створити шину з обраних пінів JTAG-мікросхеми або кластеру.
Приклад:
DeclareGroup("D301.ADDRESS",["D301.A15","D301.A14","D301.A13","D301.A12","D301.A11","D301.A10","D301.A9","D301.A8","D301.A7","D301.A6","D301.A5","D301.A4","D301.A3","D301.A2","D301.A1","D301.A0"]).
- **GetGroup(Grpname)** – дозволяє зчитувати дані зі створеної раніше шини.
Приклад: a = GetGroup("Decoder.OUTPUTS").

3.14 Використання AutoBuzz в Script

Загальна тестова послідовність має вигляд:



Рисунок 3.24 – Загальна тестова послідовність

Стандартні апаратні засоби включають в себе:

- Контролер JTAG Live.
- Контролер JT3705/USB.

- Модуль JT2111/MPV.
- Модуль JT5112.

```

1 from jft import *
2 #import jftautobuzz
3 from jftautobuzz import *
4 #*****#
5 # Program
6 #*****#
7 start("jt2156.csv")
8 extra = getExtraConnections()
9 print ("Detected extra-connections:", extra)

```

```

Live\projects\training\Test_JT2156\Test_JT2156\Test_JT2156.py

Initializing Jft...
Detected extra-connections: ['U4.J17 -> U4.H17', 'U4.H17 -> U4.J17']
Closing Jft...
Script ended successfully.

```

Рисунок 3.25 – Вікно робочого додатку з використанням AutoBuzz

3.15 Стратегія вибору точок для модифікації комбінованої схеми

Стратегія вибору точок для модифікації комбінованої схеми полягає в наступному: вибирають 3% ліній з мінімальними значеннями SU^0 (окрім зовнішніх входів і виходів), при цьому до вибраних ліній додаються ще й ті, які мають значення показника, що дорівнює максимальному, з 3% вибраних, якщо такі є. Зазвичай ліній з однаковими значеннями показників порівняно мало - це особливість методу оцінки. Теж саме робиться з значеннями SU^1 і OY . Отримані множини точок об'єднуються.

Спосіб модифікації комбінованої схеми: на кожну лінію схеми ставиться комірка, яка заздалегідь забезпечує 100% керуємість (спостерігаємість) обраної лінії, при цьому вона повинна бути абсолютно прозорою для нормального режиму роботи схеми, а також повинна просто тестуватися. Для зменшення кількості додаткових входів і виходів комірки об'єднують у зсувний регістр, аналогічний шляху сканування в структурних методах тестопридатного проектування. Комірка регістра граничного сканування (IEEE 1149.1 Boundary Scan standard) [20]) цілком задовольняє переліченим вимогам, за винятком її розмірів.

Стратегія модифікації синхронних послідовних схем: на кожному з ліній всіх локальних і глобальних зворотних зв'язків ставиться комірка сканування для комбінаційної схеми.

Стратегія модифікацій асинхронних послідовних схем: всі тригери видаляються зі схеми, в результаті отримується комбінаційна схема. Далі для отриманої таким чином комбінаційної схеми розраховуються показники тестопридатності.

3.16 Використання технології JTAG

Стандарт допускає введення додаткових режимів роботи механізму граничного сканування, тому для здійснення налаштувальних робіт фірми-розробники електронних компонентів спеціально вводять нові інструментальні режими, розширюючи систему команд та формати даних інтерфейсу.

Найважливішими складовими інструментального забезпечення механізму JTAG вбудованих обчислювальних систем є:

1. засоби опису ресурсів механізму граничного сканування;
2. інструментальні програмно-апаратні крос-засоби;
3. інструментальні резидентні програмні засоби.

3.16.1 Тестування

Цей режим функціонування передбачає застосування (перевірку) цільової системи на наборі тестових шаблонів. Набір таких шаблонів складається яким-небудь засобом і забезпечується їх виконання в системі. Зазвичай це виглядає як виставлення заданих значень в деяких точках системи, замикання реакції системи, порівняння зі зразком і так далі. Результатом може бути PASSED або FAILED. Для вирішення завдання формального опису процедури тестування використовується мова SVF. З її допомогою можна вказати послідовність зразкових векторів та векторів, що виставляються. У семантиці мови не підтримуються керуючі конструкції, можливо тільки лінійне виконання програми. В результатуючому сценарії відсутні явні зв'язки зі структурою системи, конкретними мікросхемами і так далі (хоча при складанні сценарію ця інформація була використана). Файл розрахований на роботу з повним JTAG-ланцюжком (докладніше цей термін обговорюється в наступному розділі).

3.16.2 Програмування

В рамках технології JTAG під програмуванням розуміють процес початкової ініціалізації (конфігурації) тієї або іншої структури в мікросхемі. При цьому використовується спеціальний алгоритм, розроблений фірмою-виробником мікросхеми, що спирається на розширення механізму JTAG. Алгоритм програмування в більшості випадків не розкривається фірмою-виробником. JTAG (в рамках базового стандарту) в такому режимі виконує роль транспортного засобу. Прикладом такого підходу є ПЛІС фірми Altera.

У ряді випадків мікросхеми, які потребують конфігурації, не мають JTAG-інтерфейсу (наприклад, flash-пам'ять, serial eeprom). Тоді їх можна програмувати, використовуючи JTAG-інтерфейс пов'язаних з ними мікросхем (рис. 3.25).

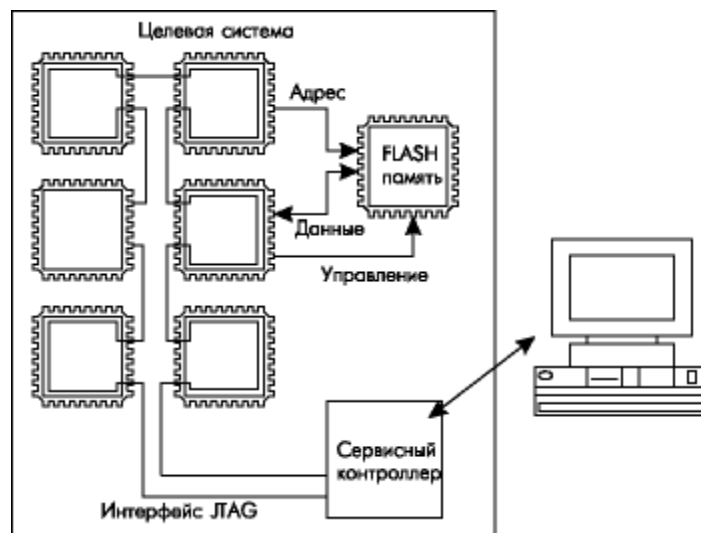


Рисунок 3.26– Типовий варіант JTAG-схеми

Алгоритми рішення таких задач складно, а часом неможливо, описати в форматі SVF. Для цього в більшій мірі підходить STAPL, мова рівня BASIC для управління TAP. У плані можливостей, STAPL характеризується примітивним управлінням, деякою процедурністю і відсутністю розвинуеного зв'язку із зовнішнім світом. Вона дозволяє безпосередньо управляти JTAG-інтерфейсом і розрядами програмованих портів введення / виводу. Мова не містить засобів роботи з файлами і не дозволяє працювати з консоллю, що обмежує її застосування. Всі дані для програмування задаються прямо в вигляді констант в тілі програми. Приклади опису мовами BSDL, SVF та STAPL наведено в додатку E.

3.16.3 Використання інструментарію

Вирішуючи задачу схемотехнічного проектування вбудованої системи, проектувальники закладають засоби початкового тестування та внутрішньо схемної ініціалізації. Для цього розробляється спеціалізований пристрій або сервісний механізм інструментально-технологічного характеру. Функціональність і склад інструментального забезпечення визначаються особливостями проекту.

Інструментальні крос-засоби являють собою сукупність програмних засобів розробки і апаратних інтерфейсів, які забезпечують доступ до цільового об'єкту з інструментальної машини. Ресурсів інструментальної машини зазвичай вистачає для реалізації інтерпретатора мовного опису механізму граничного сканування при проведенні більшості робіт з макетним зразком системи на етапах часткової ініціалізації і початкової налагодження.

Ефективність використання технології граничного сканування багато в чому залежить від вирішення двох питань: опису структури ланцюжка JTAG і формулювання алгоритму роботи з ним. Під JTAG-ланцюжком (scan path) розуміється повна стандартна тест-шина, отримана послідовним з'єднанням сигналів TDI і TDO декількох компонентів. Це поняття використовується при вирішенні наступних завдань:

- опису окремих мікросхем, що підтримують механізм граничного сканування;
- опису структури цільової системи з точки зору механізму граничного сканування;
- опису (виділення) цілі роботи конкретного алгоритму (програмування, тестування).

Зазвичай в реальній системі є кілька мікросхем, пов'язаних в один JTAG-ланцюжок. Для опису ланцюжків і їх ієрархії використовуються мови BSDL і HSDL. Як засоби опису вони ефективні, але практично не підтримані доступними інструментальними засобами. В результаті розробник задає ланцюжок JTAG-пристроїв тим чи іншим неформальним чином, а це призводить до різкого зростання трудомісткості при використанні механізму граничного сканування.

3.17 Висновки до розділу 3.

В даному розділі було досліджено складові частину JTAG-інтерфейсу, структуру граничного сканування методом JTAG . Розглянуто задачі та можливі реалізації обраного методу. Оглянуто мови програмування для опису задач тестування даного стандарту. Наголошено на стратегії вибору точок для модифікації комбінаційної схеми.

Запропоновано метод підвищення якості тесту на основі методу граничного сканування Boundary Scan. На основі проведеного дослідження отримані наступні результати: механізм граничного сканування дозволяє успішно вирішувати задачі програмування, налаштування, тестування в системах різної складності. Метод JTAG може бути використаним для вирішення широкого спектру задач, а завдяки різним додаткам та постійному додаванню функцій розробниками схем та елементів, спектр вирішуваних задач та можливих реалізацій методу граничного сканування з використанням JTAG, що сьогодні вже стало синонімічним, постійно поповнюється. Широке використання даний стандарт отримав завдяки поєднанню в собі можливостей інших методів вбудованого тестування цифрових схем. Перелічимо деякі з можливостей стандарту тестування:

- діагностика проекту з метою виявлення технологічних дефектів;
- конфігурація програмованих пристроїв (мікроконтролери, ПЛІС);
- конфігурація мікросхем пам'яті (FLASH, EEPROM);
- робота з механізмами самоконтролю;
- можливість фрагментарною налагодження компонентів;

реалізація режимів внутрішньосхемною налаштування та моніторингу, OCD (On Chip Debug);

вирішення задачі функціональної верифікації (ASIC або механізмів, що реалізуються на ПЛІС).

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даного проекту магістра було дослідження методів вбудованого тестування цифрових апаратів. Так як процес розробки виконувався у домашніх умовах, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для приміщення, де проводились роботи над дипломним проектом.

4.1 Аналіз потенційно небезпечних і шкідливих виробничих факторів, що впливають на персонал

У дипломному проекті досліджуються методи вбудованого тестування цифрових апаратів. Розроблюваний проект орієнтований на роботу з персональним комп'ютером. Використовувані для вирішення завдань ПЕОМ типу IBM PC мають такі характеристики:

- споживана потужність 220 Вт;
- робоча напруга 220 В;
- напруга джерел живлення +12 В; - 12 В, 5 В.

На користувачів під час роботи з комп'ютерною технікою можуть діяти такі види небезпек:

- ураження електричним струмом;
- енергетична небезпека (виникає через коротке замикання: опіки, електрична дуга, викид розплавленого металу);
- небезпека загоряння; термонебезпека (дія високих температур через нагрівання конструктивних елементів);
- механічна небезпека (травми через падіння, дію рухомих частин, поріз за гострі частини конструктивних елементів);
- небезпека випромінювання (дія звукового (акустичного), високочастотного, інфрачервоного, ультрафіолетового й іонізуючого випромінювання, а також видимого світла когерентної високої інтенсивності (лазерного випромінювання);

- хімічна небезпека (контакт із деякими хімікатами, які використовують для того, щоб обслуговувати обладнання, або від вдихання їх парів).

При обслуговуванні ПЕОМ мають місце фізичні та психофізичні небезпеки, а також шкідливі виробничі фактори:

- підвищений значення напруги в електричному ланцюзі, замикання якого може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювання;
- підвищений або знижена температура повітря робочої зони;
- підвищений або знижена рухливість повітря;
- підвищений або знижена вологість повітря;
- відсутність або нестача природного світла;
- підвищений пульсація світлового потоку;
- недостатність освітлення робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційне навантаження;
- монотонність праці.

4.2 Заходи з охорони праці

Основним небезпечним фактором при роботі з ЕОМ є небезпека ураження людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на обладнанні.

Проходячи через тіло людини, електричний струм чинить на нього складний вплив, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон та інших органів тканин організму) дій.

Тяжкість ураження людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає такі технічні засоби і засоби, що попереджають людини від ураження електричним струмом:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне поділ мереж;
- використання малої напруги;
- ізоляція частин, які проводять струм;
- огороження електроустановок.

Занулення зменшує напругу дотику і обмежує час, протягом якого людина, доторкнувшись до корпусу, може потрапити під дію напруги.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Трудова діяльність людини завжди протікає в певних метеорологічних умовах, які визначаються поєднанням температури повітря, швидкості його руху і відносної вологості, тиском і тепловим випромінюванням від нагрітих поверхонь. Оскільки експлуатація проектного програмного засобу відбувається в приміщенні, то ці показники в сукупності (за винятком тиску) називаються мікрокліматом виробничого приміщення. В даний час основним нормативним документом нормалізації мікроклімату є ДСН 3.3.6.042-99 «Санитарные нормы микроклимата производственных помещений [6]. Тяжкість праці характеризує сукупну дію всіх елементів, що складають умови праці, на працездатність людини, його здоров'я, життєдіяльність і відновлення робочої сили. У такому представлені поняття тяжкості праці однаково застосовне як до розумової, так і до фізичної праці. Відповідно з ДСН 3.3.6.042-99 «Санитарные нормы микроклимата производственных помещений [6] тяжкість роботи персоналу, який обслуговує ЕОМ, відноситься до легкої категорії 1а (роботи, що виконуються сидячи, не вимагаючи систематичного фізичного напруження і перенесення важкостей) [7]. Оптимальні норми мікроклімату в робочій зоні, забезпечувані для робіт легкої категорії 1а приведені в таблиці 4.1

Таблиця 4.1 – Оптимальні норми мікроклімату

Період року	Температура, °С	Відносна вологість, %	Швидкість руху повітря, м/с, не більш
Холодний	22 – 24	60 – 40	0,1
Теплий	23 – 25	60 – 40	0,1

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти) і установки кондиціонера. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в ДБН (30 кубічних метрів на годину на одного працюючого).

Для захисту від електромагнітного випромінювання передбачаються наступні заходи:

- застосування нових плазмових моніторів;
- віддалення робочого місця не менше, ніж на 0,4 – 0,5 м, оскільки напруженість електричного поля зменшується при віддаленні від джерела поля;
- встановлення раціональних режимів роботи персоналу (обмеження часу перебування);
- раціональне розміщення в робочому приміщенні устаткування, що випромінює електромагнітну енергію.

Оскільки рівень шуму не перевищує гранично допустимих величин [8, 9], які встановлені санітарними нормами, заходи для зниження шуму не проводяться.

Для зниження стомлюваності обслуговуючого персоналу в приміщеннях, де розташовані обчислювальні засоби, передбачається використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

У проекті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників N виробляється по формулі (4.1):

$$N = \frac{E \cdot l \cdot m \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.1)$$

Де E – нормована освітленість – 200 лк;

l – довжина кімнати – 5 м;

m – ширина кімнати – 3 м;

Z – поправочний коефіцієнт світильника (для стандартних світильників $Z = 1.1 - 1.3$) приймаємо рівним 1,2;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,55

M – число люмінесцентних ламп в світильнику – 1;

F – світловий потік лампи – 5400 лм.

Підставивши числові значення у формулу (4.1), отримуємо:

$$N = \frac{200 \cdot 5 \cdot 3 \cdot 1,2 \cdot 1,5}{5400 \cdot 0,55 \cdot 1} = 1,8$$

Згідно розрахунку приймаємо два світильника по одній лампі.

4.4 Рекомендації з пожежної профілактики

Пожежі в робочому приміщенні становлять небезпеку, тому що пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки.

Пожежа може виникати при внесенні джерела запалювання в горючу середу. Горючими матеріалами в приміщенні, де розташовані обчислювальні засоби є будівельні матеріали, віконні рами, двері, підлоги, меблі, ізоляція силових і сигнальних кабелів, радіотехнічні деталі, конструктивні елементи з пластичних матеріалів, рідини для очищення елементів і вузлів ЕОМ від забруднень:

- 1) поліамід – матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420 °С,
- 2) полівінілхлорид – ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С,
- 3) стеклотекстоліт ДЦ – матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.74, не схильний до температурного самозаймання,
- 4) пластикат кабельний №.489 – матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1,
- 5) деревина – будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255 °С, температура самозаймання 399 °С.

Згідно ДСТУ Б В.1.1-36:2016 Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою [7] таке приміщення відноситься до категорії "В" (пожежонебезпечної) .

Пожежа може виникнути в результаті утворення джерела запалювання (іскри і дуги короткого замикання, порушення ізоляції, що призводить до короткого замикання, перегріву радіодеталей внаслідок тривалого перевантаження) і внесення його в горючу середу.

При повному згорянні органічних сполук утворюється (CO_2 , SO_2 , H_2O , N_2), а при згорянні неорганічних сполук - оксиди. Залежно від температури плавлення продукції, реакції диму можуть знаходитися у вигляді розплаву (Al_2O_3 , TiO_2), або підніматися в повітря у вигляді диму (P_2O_5 , Na_2O , MgO). Розплавлені тверді частинки створюють світність полум'я. Склад продуктів неповного згорання горючих речовин складний і різноманітний. Це можуть бути горючі речовини - H_2 , CO , CH_4 та інші; атомарний водень і кисень; різні радикали - OH , CN та інші. Продуктами неповного згорання можуть бути також оксиди азоту, спирти альдегіди, кетони і високотоксичні з'єднання, наприклад, синильна кислота.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза, що фільтрує, з коробкою марки В (жовтий).

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем вентиляції і кондиціонування, розвиненою системою електроживлення ЕОМ. Небезпека загорання в ЕОМ пов'язана із значною кількістю щільно розташованих на монтажній платі і блоках електронних вузлів і схем, електричних і

комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80...100 °С), що може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання.

Пожежна безпека при застосуванні ЕОМ забезпечується:

- 1) системою запобігання пожежі;
- 2) системою протипожежного захисту;
- 3) організаційно-технічними заходами.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важкогорючі) не надається технічно можливим. Тому проектом передбачаються способи і засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання, таких як:

- 1) застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонам відповідно до ПУЕ;
- 2) застосування в конструкції швидкодійних засобів захисного відключення можливих джерел запалення;
- 3) виключення можливості появи іскрового розряду в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення.

Для протипожежного захисту проектом передбачається використання автоматичну пожежну сигналізацію із застосуванням датчика-сповіщувача РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт. і застосуванням первинних засобів пожежогасіння. Площа контрольована сповіщувачем 150 м². Відповідно до норм первинних засобів пожежогасіння пропонується використовувати:

- ручний вуглекислий вогнегасник ОУ-5 в кількості 1 шт.
- повсть 1×1 м², кошму 2×1,5 м² або азбестове полотно 2×2 м² в кількості 1 шт.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу на тему пожежної безпеки.

4.5 Розрахунок захисного заземлення

Розрахунок проводять за допомогою методу коефіцієнта використання електродів. Коефіцієнт використання вертикальних заземлювачів η_v в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію

горизонтального заземлювача враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c .

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з}$:

$$R_{шт.з} = (R_d \cdot R_{пр.з}) / (R_{пр.з} - R_d) \quad (4.3)$$

де $R_{пр.з}$ – опір природних заземлювачів; R_d – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з} = R_d$.

Підставивши числові значення у формулу (4.3), отримуємо:

$$R_{шт.з} = (4 \cdot 40) / (40 - 4) = 4 \text{ Ом} \quad (4.4)$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом·м. Приблизне значення питомого опору глини приймаємо $\rho = 40$ Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту, $\rho_{розр}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в}$, і горизонтальних $\rho_{розр.г}$, Ом·м за формулою:

$$\rho_{розр} = \psi \cdot \rho \quad (4.5)$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{розр.в} = 1,7$ і горизонтальних $\rho_{розр.г} = 5,5$ Ом·м.

$$\rho_{розр.в} = 1,7 \cdot 40 = 68 \text{ Ом} \cdot \text{м}$$

$$\rho_{розр.г} = 5,5 \cdot 40 = 220 \text{ Ом} \cdot \text{м}$$

4) Розраховується опір розтікання струму вертикального заземлювача R_v , Ом,

$$R_v = (\rho_{розр.в}) / (2 \cdot \pi \cdot l_v) \cdot (\ln(((2 \cdot l_v) / (d_{ст})) + 1) / 2) \cdot \ln((4 \cdot t + l_v) / (4 \cdot t - l_v))) \quad (4.6)$$

де l_v – довжина вертикального заземлювача (для труб - 2–3 м; $l_v = 3$ м);

$d_{ст}$ – діаметр стержня (для труб - 0,03–0,05 м; $d_{ст} = 0,05$ м);

t – відстань від поверхні землі до середини заземлювача

$$R_B = (68) \cdot (2 \cdot \pi \cdot 13) \cdot (\ln((2 \cdot 3) \cdot (0,05)) + 1/2) \cdot \ln((4 \cdot 2,3 + 3) \cdot (4 \cdot 2,3 - 3)) = 18,5 \quad (4.7)$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання η_v :

$$\eta_v = (2 \cdot R_B) / R_d = 9,25 \quad (4.8)$$

Γ визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_v = 0,57$ (табличне значення).

6) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання n_v , шт:

$$\eta = (2 \cdot R_B) / (R_d \cdot \eta_v) \approx 16 \quad (4.9)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 \times L_B \times (n_v - 1) \quad (4.10)$$

де L_B – відстань між вертикальними заземлювачами, (прийняти за $L_B = 3$ м);
 n_v – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \times 3 \times (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) R_r , Ом:

$$R_r = \rho_{\text{розр.в}} \cdot (2 \cdot \pi \cdot l_c) \cdot \ln((2 \cdot l_c^2) / (d_{\text{см}} \cdot h_r)) \quad (4.11)$$

де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15$ м;
 h_r – глибина закладання горизонтальних заземлювачів (0,5 м);
 l_c – довжина з'єднувальної стрічки горизонтального заземлювача l_c , м

$$R_r = 220 \cdot (2 \cdot \pi \cdot 48) \cdot \ln((2 \cdot 48^2) / (0,95 \cdot 0,15 \cdot 0,5)) = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача η_c відповідно до необхідної кількості вертикальних заземлювачів пв.

Коефіцієнт використання з'єднувальної смуги $\eta_c=0,3$ (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{заг} = (R_v \cdot R_r) / (R_v \cdot \eta_c + R_r \cdot \eta_v) \leq R_d \quad (4.12)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{заг} < 4 \text{ Ом}$, а саме:

$$R_{заг} = (18,5 \cdot 8,1) / (18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57) = 1,9 \leq R_d$$

4.6 Охорона навколишнього природного середовища

Діяльність за темою магістерської роботи в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства [12-14].

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на видалення (знешкодження), утилізацію, тощо в ІТ галузі.

Немає впливу на атмосферне повітря при нормальних умовах праці, бо в приміщенні не використовуються сканери, принтери та інші джерела викиду забруднюючих речовин в повітря робочої зони.

В процесі діяльності користувача виникають процеси поводження з відходами ІТ галузі. Види відходів, утворення, яких можливо:

- відпрацьовані люмінесцентні лампи - I клас небезпеки;
- батарейки та акумулятори (малі) - III клас небезпеки;
- змінні носії інформації - IV клас небезпеки;
- відпрацьований ізолюючий матеріал, дроти та кабелі - IV клас небезпеки;
- макулатура - IV клас небезпеки;
- побутові відходи - IV клас небезпеки.

4.7 Перелік джерел посилань до розділу 4

1. ДБН В.2.5-28:2015 «Природне і штучне освітлення». Режим доступу: https://dbn.co.ua/load/normativy/dbn/dbn_v_2_5_28/1-1-0-1188
2. НАПБ А.01.001-2004 «Правила пожежної безпеки України». Режим доступу: <https://zakon.rada.gov.ua/laws/show/z1410-04> 04.11.2004
3. ДБН В.2.5-67:2013. «Опалення, вентиляція та кондиціонування». Режим доступу: <https://dbn.co.ua/load/normativy/dbn/1-1-0-1018> 01.01.14
4. ГОСТ 13109-97 «Електрична енергія. Сумісність технічних засобів. Норми якості електричної енергії в системах електропостачання загального призначення». Режим доступу: https://dnaop.com/html/42313/doc-0%D0%93%D0%9E%D0%A1%D0%A2_13109-97
5. Правила технической эксплуатации электроустановок потребителей, утвержденные приказом Минтопливэнерго Украины от 25.07.2006 № 258 (в редакции приказа от 13.02.2012 № 91), которыми унормированы организационные и технические требования к эксплуатации электроустановок потребителей. Режим доступу: http://search.ligazakon.ua/l_doc2.nsf/link1/RE13017.html
6. ДСН 3.3.6.042-99 «Санитарные нормы микроклимата производственных помещений». Режим доступу: https://dnaop.com/html/31678/doc-%D0%94%D0%A1%D0%9D_3.3.6.042-99 01.12.99
7. ДСН 3.3.6.037-99. Санітарні норми виробничого шуму, ультразвуку та інфразвуку. Режим доступу: <https://zakon.rada.gov.ua/rada/show/va037282-99> 01.12.99
8. Наказ МСПУ №207 “Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями”. Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18> 25.04.2018
9. ДЕРЖАВНІ САНІТАРНІ НОРМИ ТА ПРАВИЛА «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу». Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0472-14> 08.04.14
10. Закон України «Про охорону навколишнього природного середовища». Режим доступу: <https://zakon.rada.gov.ua/laws/show/1264-12> 12.10.18
11. Закон України «Про забезпечення санітарного та епідемічного благополуччя населення». Режим доступу: <https://zakon.rada.gov.ua/laws/show/4004-12> 04.10.18
12. Закон України «Про відходи». Режим доступу: <https://zakon.rada.gov.ua/laws/show/187/98-%D0%B2%D1%80> 01.05.19

13. Закон України «Про охорону атмосферного повітря». Режим доступу:
<https://zakon.rada.gov.ua/laws/show/2707-12> 18.12.17
14. Закон України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру». Режим доступу:
<https://zakon.rada.gov.ua/laws/show/1809-14> 02.10.12
15. ДСТУ Б В.1.1-36:2016 Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою. Режим доступу:
https://dbn.co.ua/load/normativy/dstu/dstu_b_v_1_1_36/5-1-0-1759 01.01.17
СТУ Б В.1.1-36:2016 Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою. Режим доступу:
https://dbn.co.ua/load/normativy/dstu/dstu_b_v_1_1_36/5-1-0-1759 01.01.17

ВИСНОВКИ

У вступі обґрунтована актуальність та важливість методів вбудованого тестування цифрових апаратів та доцільність даної дослідницької роботи.

У першому розділі магістерської роботи проведено дослідження існуючих методів вбудованого тестування ВІС. Розглянуті питання та проблеми, що є типовими для обраної теми. Визначено переваги обраного метода для подальшого дослідження.

У другому розділі магістерської роботи проведено дослідження проектування систем на кристалі, визначено та досліджено типові елементи систем, що мають в собі вбудовані методи тестування. Досліджено та розглянуто типові моделі несправностей при тестуванні та налаштуванні цифрових схем та апаратів. Досліджено принципи роботи основних логічних елементів. Визначено основні проблеми даної галузі.

В третьому розділі магістерської роботи проведено дослідження граничного сканування методом JTAG. Розглянуто ключові елементи обраного методу. Розглянуто задачі та можливі реалізації обраного методу. Оглянуто мови програмування для опису задач тестування даного стандарту. Наголошено на стратегії вибору точок для модифікації комбінаційної схеми. Запропоновано метод підвищення якості тесту на основі методу граничного сканування Boundary Scan. На основі проведеного дослідження отримані наступні результати: механізм граничного сканування дозволяє успішно вирішувати задачі програмування, налаштування, тестування в системах різної складності. Метод JTAG може бути використаним для вирішення широкого спектру задач, а завдяки різним додаткам та постійному додаванню функцій розробниками схем та елементів, спектр вирішуваних задач та можливих реалізацій методу граничного сканування з використанням JTAG, що сьогодні вже стало синонімічним, постійно поповнюється. Широке використання даний стандарт отримав завдяки поєднанню в собі можливостей інших методів вбудованого тестування цифрових схем. Показано вирішення типових проблем та задач. Перелічимо деякі з можливостей стандарту тестування:

1. діагностика проекту з метою виявлення технологічних дефектів;
2. конфігурація програмованих пристроїв (мікроконтролери, ПЛІС);
3. конфігурація мікросхем пам'яті (FLASH, EEPROM);
4. робота з механізмами самоконтролю;
5. можливість фрагментарною налагодження компонентів;
6. реалізація режимів внутрішньосхемною налаштування та моніторингу, OCD (On Chip Debug);

вирішення задачі функціональної верифікації (ASIC або механізмів, що реалізуються на ПЛІС).

У четвертому розділі магістерської роботи виконаний аналіз потенційних небезпек при роботі із засобами обчислювальної техніки, на підставі якого розроблено заходи з техніки безпеки, заходи, що забезпечують виробничу санітарію та гігієну праці, рекомендації з пожежної профілактики, які підтверджені відповідними розрахунками. Проведено аналіз впливу на навколишнє природне середовище під час виконання магістерської роботи.

ПЕРЕЛІК ПОСИЛАНЬ ДО РОЗДІЛІВ 1-3

1. ГОСТ 18322-78. Система технического обслуживания и ремонта техники. Термины и определения.
2. ГОСТ 20922-75. Техническая диагностика. Термины и определения.
3. Абдуллаев Д.А., Арипов М.Н. Основы эксплуатации систем передачи дискретных сообщений. - Т.: Фан, 1984. - 113с
4. Берганов И.Р. Автоматизация технического обслуживания первичной сети связи. Т.: Изд-во Фан АН РУз. 1996. - 174с.
5. Джейкокс Дж. Руководство по поиску неисправностей в электронной аппаратуре. Перевод с англ. - М.: Мир, 1989. - 176с.
6. [Prince 1991] В. Prince, Semiconductor Memories: A Handbook of Design, Manufacture and Application, 2nd ed., John Wiley & Sons, Chichester, 1991.
7. Тестирование цифровых устройств. Режим доступа до матеріалів публікації: http://portal.tpu.ru:7777/SHARED/t/TRACEY/Courses/theory/Tab_automata_textbooks/04.pdf
8. Владимир Коробицын «Элементы и устройства вычислительной техники и информационных систем» // Москва, 2011с. 1-2
9. Матеріали лекцій відкритого освітнього ресурсу «ИНТУИТ». Лекція 1, с.1. Режим доступа до джерела: <https://www.intuit.ru/studies/courses/3440/682/lecture/14047>
10. ГОСТ 25866-83. Эксплуатация техники. Термины и определения.
11. С.-Т. Huang, J.-R. Huang, С.-F. Wu, С.-W. Wu, and Т.-Y. Chang, A programmable BIST core for embedded DRAM, IEEE Des. Test Comput., 16(1), 59-70, 1999.
12. С.-W. Wu, Testing embedded memories: is BIST the ultimate solution?, in Proc. Asian Test Symp., November 1998, pp. 516-517.
13. Скобцов В.Ю. Логическое моделирование и тестирование цифровых устройств / В.Ю Скобцов., Ю.А.Скобцов. – Донецк: ИПММ НАНУ, ДонНТУ, – 2005. – 436с.
14. Ю. Ф. Адамов. Проектирование систем на кристалле, Москва 2005, 112с
15. Дербунович Л.В., Темников И.Н., Татаренко Д.А. Генераторы тестов для дискретных устройств с самотестированием // Информационно-управляющие системы на железнодорожном транспорте. – 2004. - №1. – С.42-46.

16. Chatterjee M., Pradhan D.K. A BIST pattern generator design for near-perfect fault coverage // *IEEE Trans of Computers*. - 2003. - Vol. 52. - № 12. - P.1543-1557.
17. Pomeranz I., Reddy S.M. A storage-Based Built-in-Self-Test Pattern Generation Method for Scan Circuits Based on Partitioning and Reduction of a Precomputed Test Set // *IEEE Trans of Computers*. -2002. - Vol. 51. - № 11. - P. 1282-1293.
18. Pomeranz I., Reddy SM. Procedure for static Compaction of test sequences for synchronous sequential circuits // *IEEE Trans of Computers*. - 2000. - Vol. 49. - № 6. - P. 596-607.
19. Hahanov V., Hyduke S. Topological BDP faultsimulation method. Proceeding of EuroMicro Symposium. France, Rennes: INRIA. 2004. P. 356-359.
20. Городецкий А. Снова о внутрисхемном тестировании ICT. Часть 1 // *Компоненты и технологии*, 2011. № 7. С. 58–59.
21. Городецкий А. Снова о внутрисхемном тестировании ICT. Часть 2 // *Компоненты и технологии*, 2011. № 8. С. 44–45.
22. Городецкий А. Снова о внутрисхемном тестировании ICT. Часть 3 // *Компоненты и технологии*, 2011. № 9. С. 6–7.
23. Albee A. J. The evolution of ICT: PCB technologies, test philosophies, and manufacturing business models are driving in-Circuit test evolution and innovations // *IPC APEX EXPO Conference and Exhibition 2013*, 1. P. 381– 401.
24. Holtzer M. In-circuit pin testing: An excellent potential source of value creation // *SMT Surface Mount Technology Magazine*, 2015, 30 (6). P. 68–71.
25. Nelson R. Systems and software support PCB test // *EE: Evaluation Engineering*, 2013, 52 (2). P. 14–17. 7. IEEE Std. 1149.1 – Standard Test Access Port and Boundary-Scan Architecture. <http://grouper.ieee.org/groups/1149/1>.
26. Renbi A., Delsing J. Application of Contactless Testing to PCBs with BGAs and Open Sockets // *Journal of Electronic Testing: Theory and Applications*, 2015, 31 (4). P. 339–347.
27. Renbi A., Delsing J. Contactless Testing of Circuit Interconnects // *Journal of Electronic Testing: Theory and Applications*, 2015, 31 (3). P. 229–253.
28. Wang, R., Chakrabarty, K., Bhawmik, S. Interconnect testing and test-path scheduling for interposer-based 2.5-D ICs // *IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems*, 2015, 34 (1), art. no. 6936331. P. 136–149.

ДОДАТОК А

Приклад роботи в системі JTAG ProVision

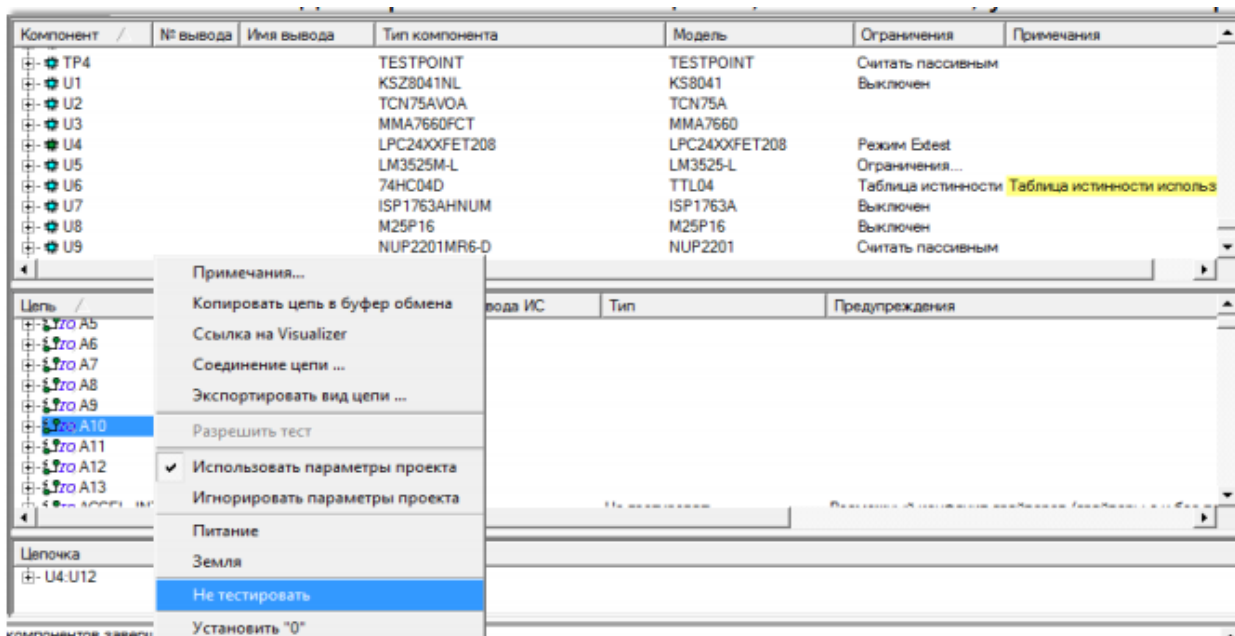


Рисунок А.1 – Редагування властивостей ланцюгів, компонентів, замовчувань

Имя цепи	Плата /	Классификация	Testability	Покрытие	Кс
U10 ALE/ADV_N	U2156_1		100%	50%	
U10 BA0	U2156_1		67%	67%	
U10 BA1	U2156_1		67%	67%	
U10 BOOT_ASDO	U2156_1		67%	67%	
U10 BOOT_ATA0	U2156_1		100%	100%	
U10 BOOT_CSO	U2156_1		100%	83%	
U10 BOOT_DCLK	U2156_1		67%	67%	
U10 BUS_A0	U2156_1		100%	50%	
U10 BUS_A1	U2156_1		100%	50%	
U10 BUS_A2	U2156_1		100%	50%	

Статистика по цепям	Тестопригодность	%	Покрытие	%
Общее количество цепей	617	100%	617	100%
Цепей в нетлисте	612		612	
Цепей, добавленных для неподключенных выводов ИС (+)	5		5	
Цепей, игнорируемых пользователем (-)	0		0	
Цепи, тестируемые периферийным сканированием	449	73%	420	68%
Считанные компонентом периферийного сканирования (прямые)	401		387	
С доступом через прозрачный компонент (непрямые)	13		2	
Цепи питания / заземления	9		8	
Цепи, тестируемые косвенно	26		23	

Рисунок А.2 – Створення звіту про тестове покриття

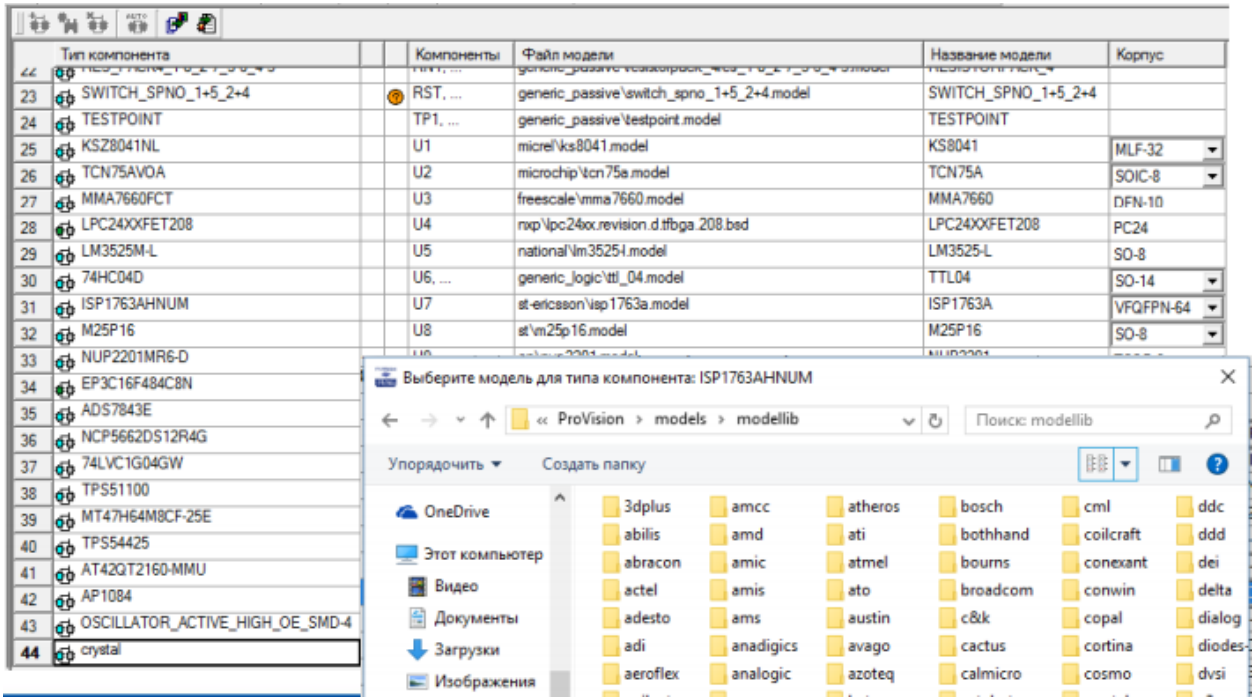


Рисунок А.3 – Менеджер компонентів та бібліотека

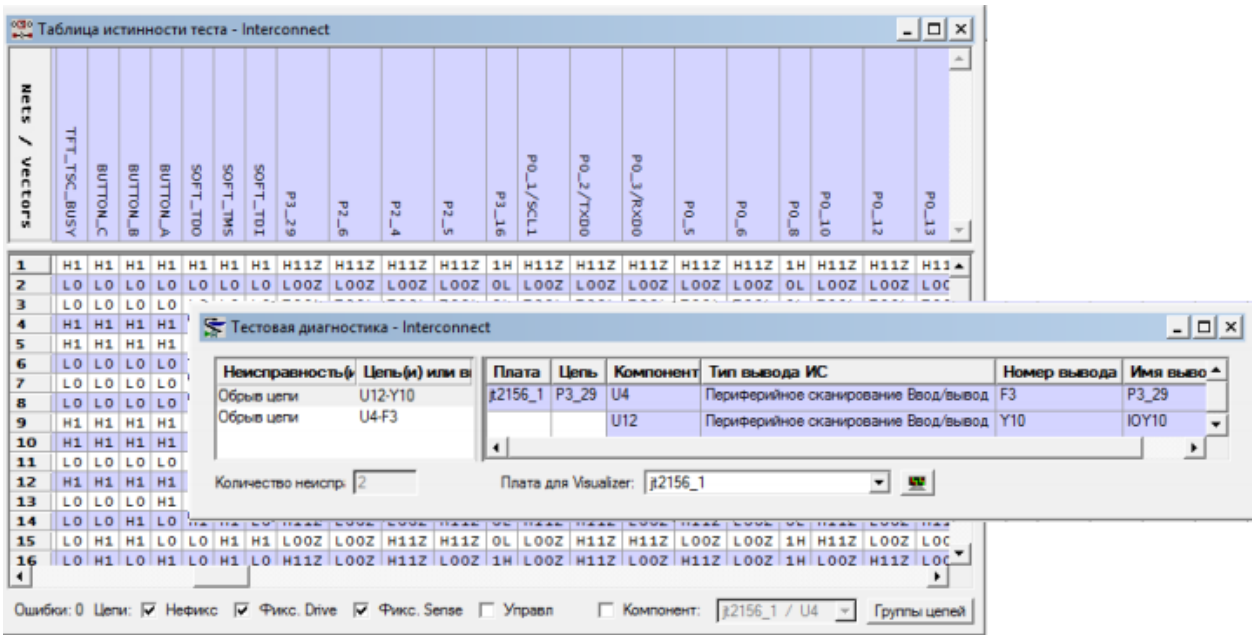


Рисунок А.4 – діалогове вікно програми після запуску тестів

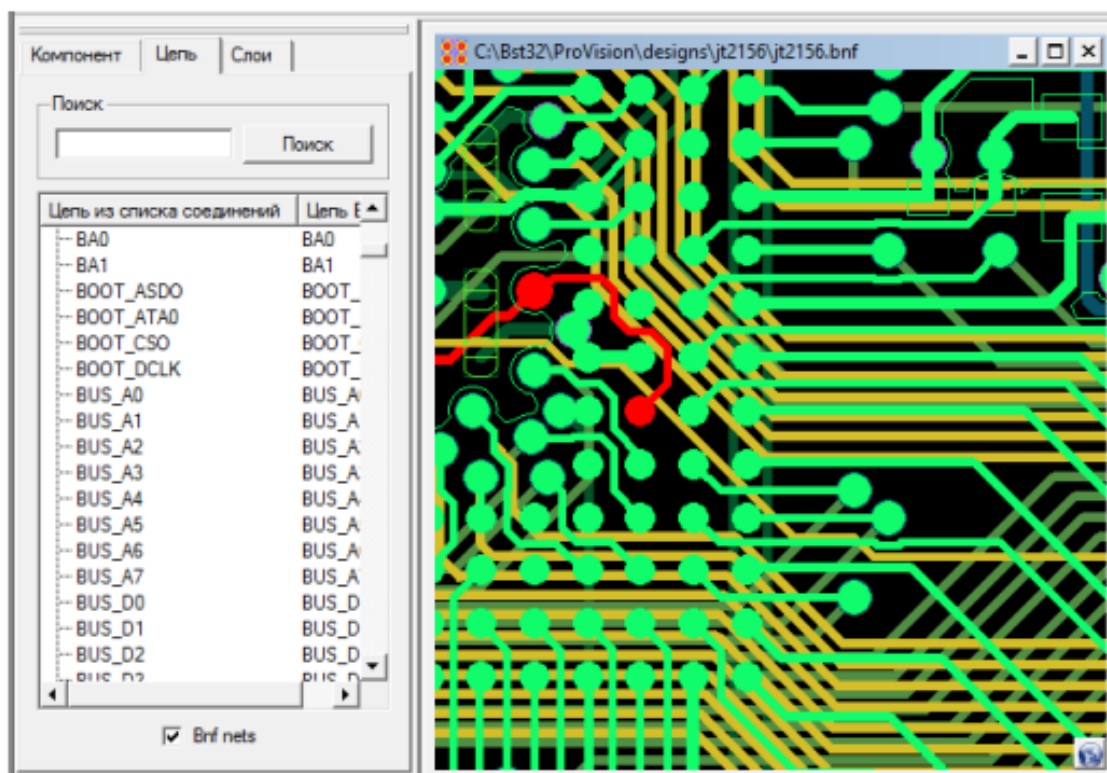


Рисунок А.5 – Диалогове вікно JTAG Visualizer після знаходження дефектів

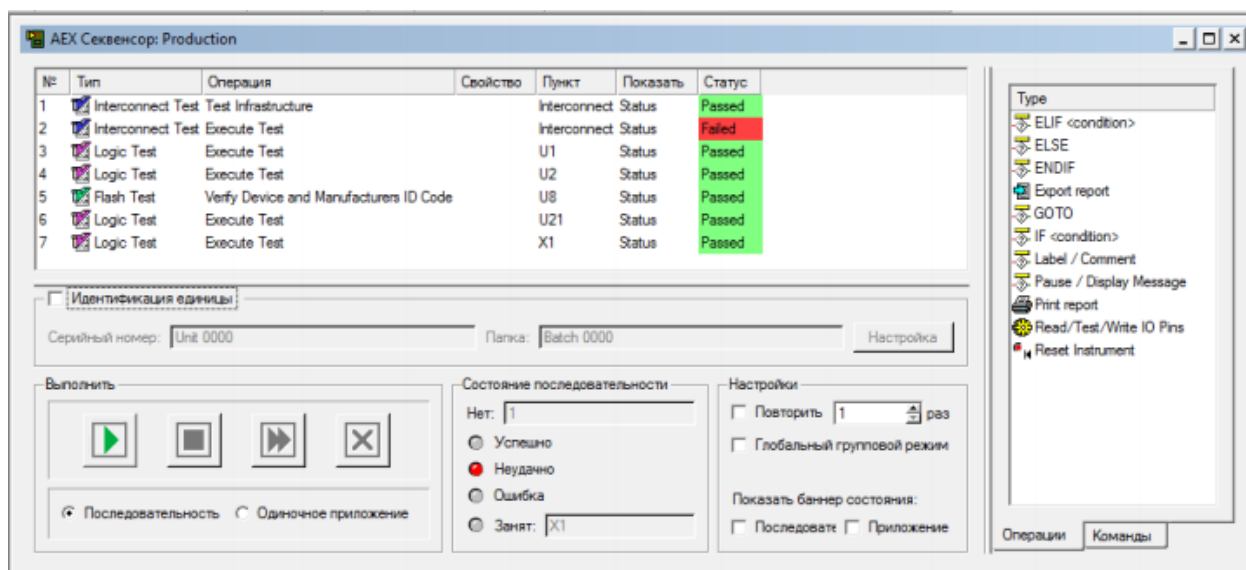


Рисунок А.6 – Секвенсор на базі платформи ProVision

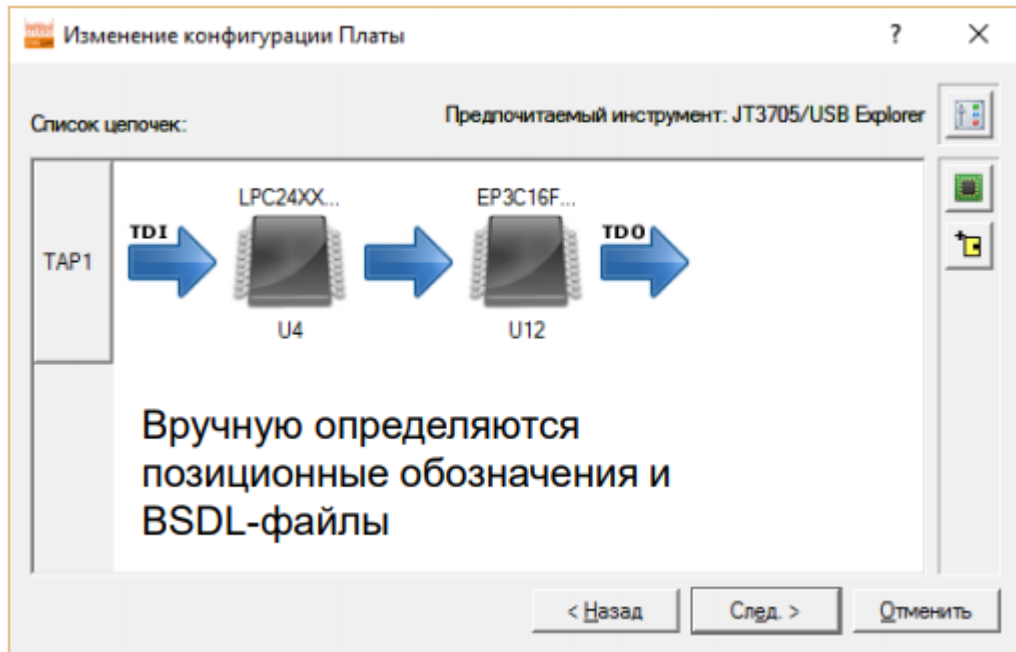


Рисунок А.7 – Автовызначення JTAG-ланцюга

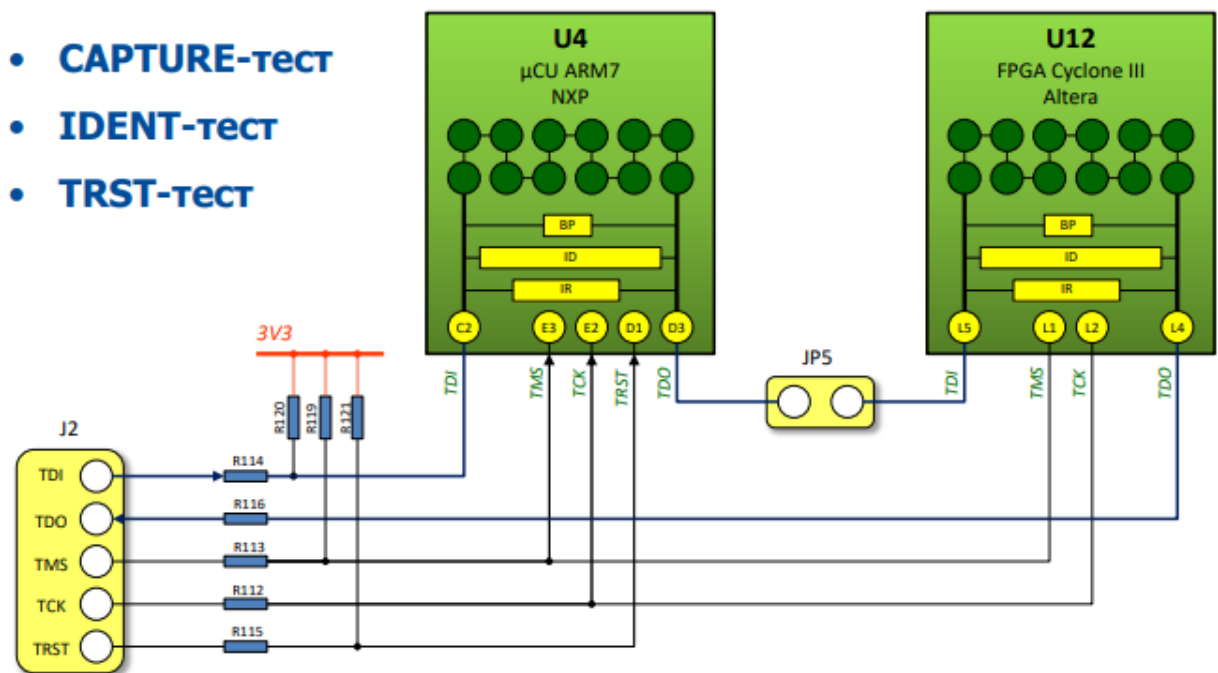


Рисунок А.8 – Тестування інфраструктури

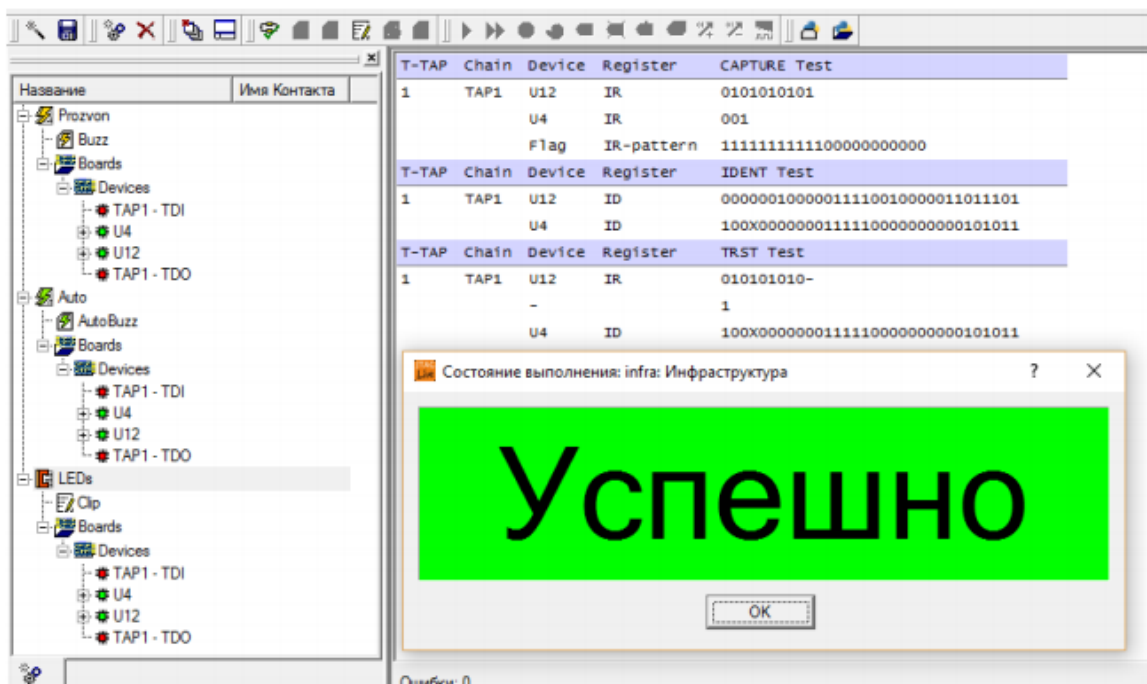


Рисунок А.9 – Результат

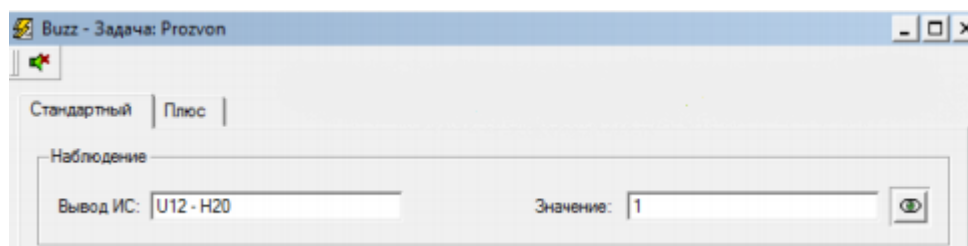


Рисунок А.10 – Моніторинг значення на піні (використовується команда SAMPLE)

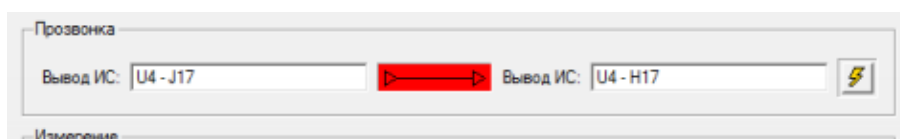


Рисунок А.11 – Продзвін в стилі мультиметра

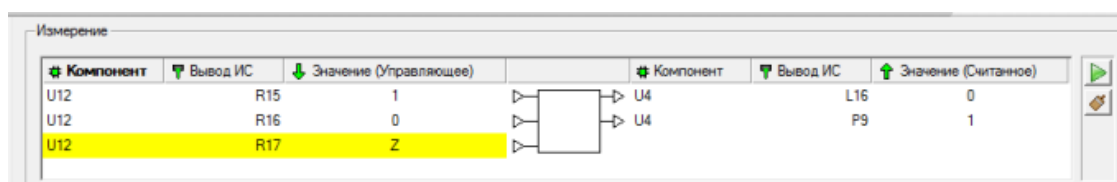


Рисунок А.12 – тестування шини в ручному режимі

ДОДАТОК Б

Приклад роботи зі схемою

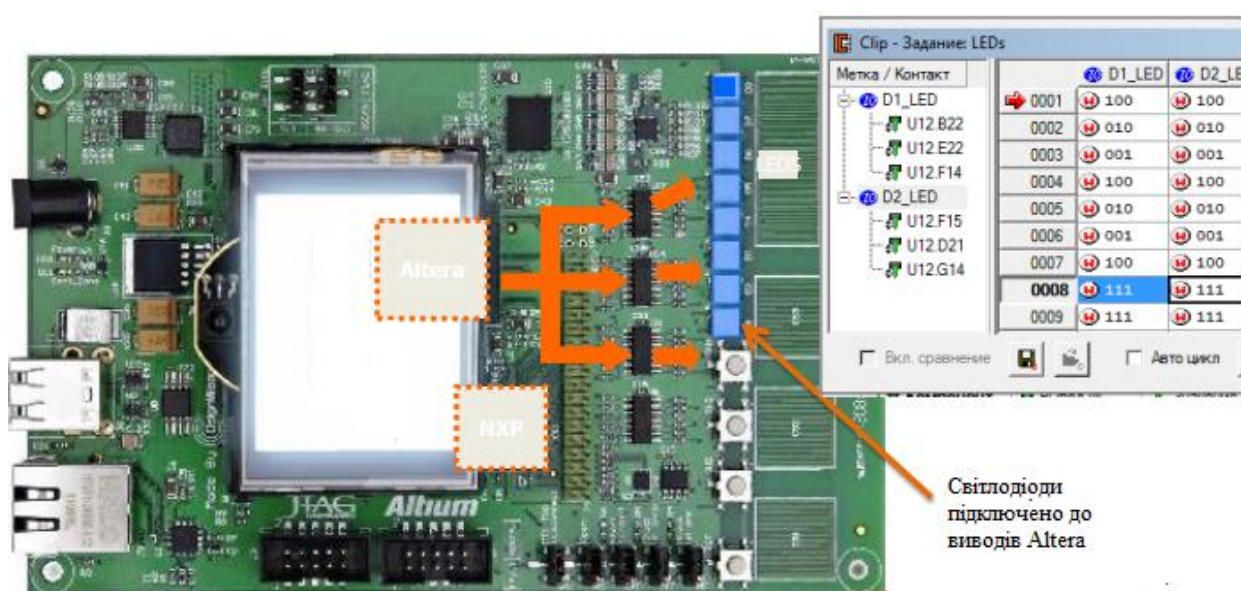


Рисунок Б.1 – Виконання «гірлянди» за допомоги CLIPS.

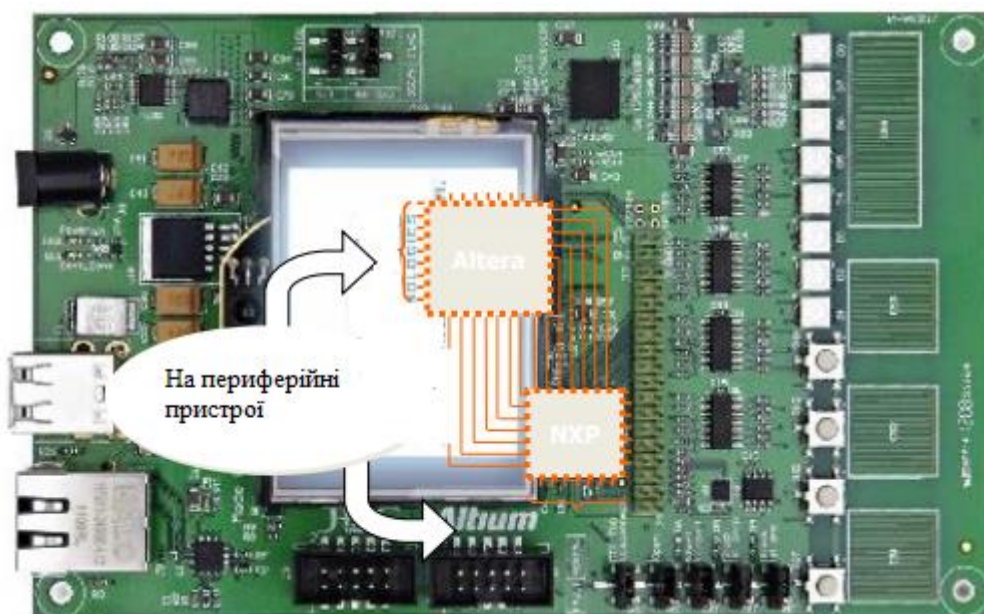


Рисунок Б.2 – Зчитування зв'язків та порівняння

ДОДАТОК В
Приклад апаратних засобів для JTAG-тестування



Рисунок В.1 - Контролер JTAG Live



Рисунок В.2 - Контролер JT3705/USB



Рисунок В.3 - Модуль JT2111/MPV



Рисунок В.4 - Модуль JT5112

ДОДАТОК Г

Діаграма станів TAP-контролеру та структура пристрою керування граничного сканування

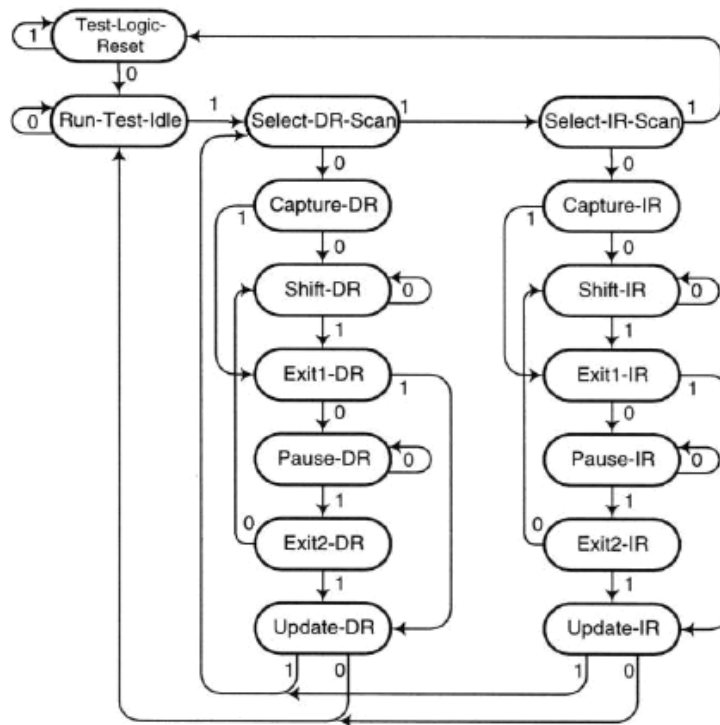


Рисунок Г.1 – Діаграма станів автомату TAP-контролера

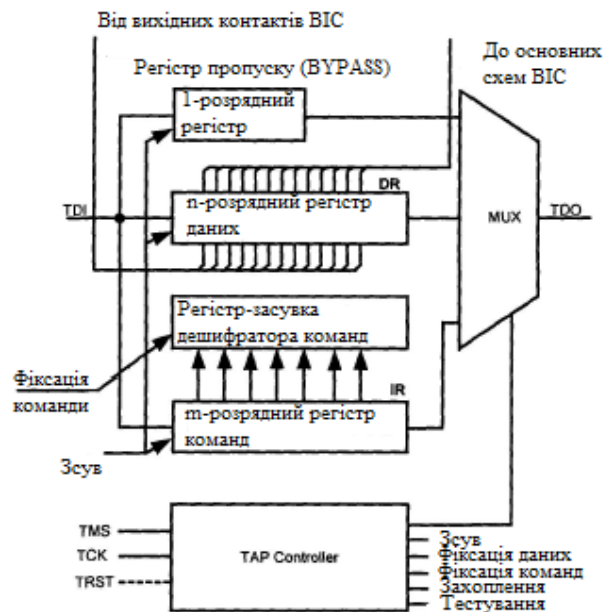


Рисунок Г.2 – Структура пристрою керування граничного сканування

ДОДАТОК Д

Приклади опису мовами програмування

```

_____ ***** ENTITY DEFINITION WITH PORTS *****
entity MY_CHIP is
  generic (PHYSICAL_PIN_MAP : string := "Undefined");
  port(
    --I/O Pins
      IO1      : in bit;
      IO1      : out bit;
      IO2      : inout bit;
    --JTAG Ports
      TCK, TMS, TDI : in bit;
      TDO      : out bit;
    --Power Pins
      VCC      : linkage bit;
    --Ground Pins
      GND      : linkage bit
  );
  use STD_1149_1_1994.all;
  attribute COMPONENT_CONFORMANCE of MY_CHIP :
    entity is "STD_1149_1_1993";
  _____ ***** PIN MAPPING *****
  attribute PIN_MAP of MY_CHIP : entity is PHYSICAL_PIN_MAP;
  constant Undefined : PIN_MAP_STRING:=
    --I/O Pins
      "IO0 : 1, IO1 : 2, IO2 : 3"&
    --JTAG ports
      "TCK : 4, TMS : 5, TDI : 6, TDO : 7, "&
    --Power Pins
      "VCC : 8, "&
    --Ground Pins
      "GND : 9 ";
  _____ *****IEEE 1149.1 TAP PORTS *****
  attribute TAP_SCAN_IN of TDI : signal is true;
  attribute TAP_SCAN_MODE of TMS : signal is true;
  attribute TAP_SCAN_OUT of TDO : signal is true;
  attribute TAP_SCAN_CLOCK of TCK : signal is (10.00e6,BOTH);
  _____ *****INSTRUCTIONS AND REGISTER ACCESS*****
  attribute INSTRUCTIONS_LENGTH of MY_CHIP : entity is 3;
  attribute INSTRUCTIONS_OPCODE of MY_CHIP : entity is
    "BYPASS (111),"&
    "EXTTEST (000),"&
    "SAMPLE (101),"&
    "IDCODE (110),"&
    "USERCODE (010),"&
  attribute INSTRUCTIONS_CAPTURE of MY_CHIP : entity is "001";

  attribute IDCODE_REGISTER of MY_CHIP : entity is
    "0000"& --4-bit Version
    "0001000000010000"& --16-bit Part Number (hex 1010)
    "000011011110"& --11-bit Manufacturer's Identity
    "1"; --Mandatory LSB
  attribute USERCODE_REGISTER of MY_CHIP : entity is
    "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"; --Bits 26-20 are programmable
  attribute REGISTER_ACCESS of MY_CHIP : entity is
    "DEVICE_ID (IDCODE)";
  _____ ***** BOUNDARY SCAN CELL INFORMATION *****
  attribute BOUNDARY_LENGTH of my_CHIP : entity is 6;
  attribute BOUNDARY_REGISTER of MY_CHIP : entity is
  -- num cell port function safe {ccell disval rslt}
    -BSC group 0 for input pin IO0
    "0 (BC_1, IO0, input, X),"&
    -BSC group 1 for output pin IO1
    "1 (BC_1, control, 1),"&
    "2 (BC_1, IO0, output3, X, 1, 1, Z),"&
    -BSC group 2 for inout pin IO2
    "3 (BC_1, IO2, input, X),"&
    "4 (BC_1, control, 1),"&
    "5 (BC_1, io2, output3, X, 4, 1, Z)";
end MY_CHIP;

```

Рисунок Д.1 – Приклад опису мовою BSDL

```

===== SVF File =====
!Begin Test Program

ENDIR IDLE; !End IR scans in IDLE
ENDDR IDLE; !End DR scans in IDLE
HIR 8 TDI (00); !8-bit IR header
HDR 16 TDI (FFFF) TDO (FFFF) MASK (FFFF); 16-bit DR header
TIR 16 TDI (0000); ! 16-bit IR trailer
TDR 8 TDI (12); ! 16-bit DR trailer
SIR 8 TDI (41); ! 8-bit IR scan
SDR 32 TDI (ABCD1234) TDO (11112222); ! 32-bit DR scan
STATE DRPAUSE; ! Go to stable DRPAUSE
RUNTEST 100 TCK ENDSTATE IRPAUSE; !RUNBIST for 100 TCKs

!End Test Program

```

Рисунок Д.2 – Пример описания на языке SVF

```

===== Stapl File =====
NOTE "Reading IDCODE from a Single Device"
ACTION READ_IDCODE = DO_READ_IDCODE;
PROCEDURE DO_READ_IDCODE;
!Declare variables for data arrays
BOOLEAN read_data[32];
BOOLEAN i_idcode[10] = #1001101000;
BOOLEAN ones_data[32] = $FFFFFFFF;
INTEGER i;
!Initialize device
STATE RESET;
!Load incode instruction
IRSCAN 10, i_idcode[9..0];
!Capture incode
DRSCAN 32 ones_data[31..0], CAPTURE read_data[31..0];
EXPORT "IDCODE", read_data[31..0];
ENDPROC;
CRC3759;

```

Рисунок Д.3 – Пример описания на языке STAPL

ДОДАТОК Е
Електронна презентація

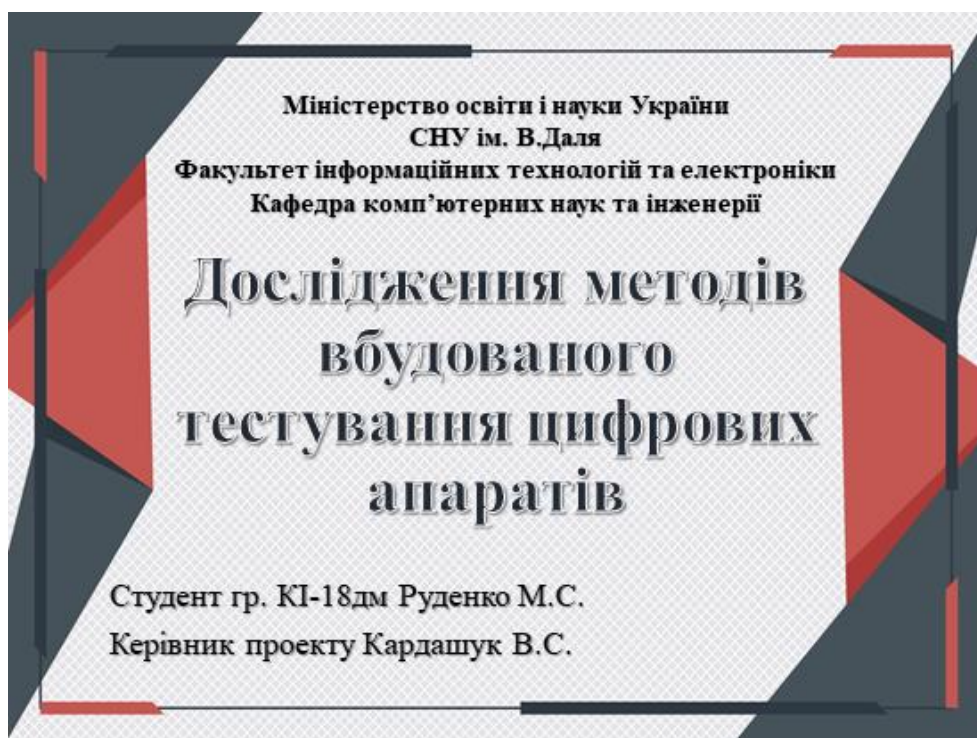


Рисунок Е.1 – Слайд 1

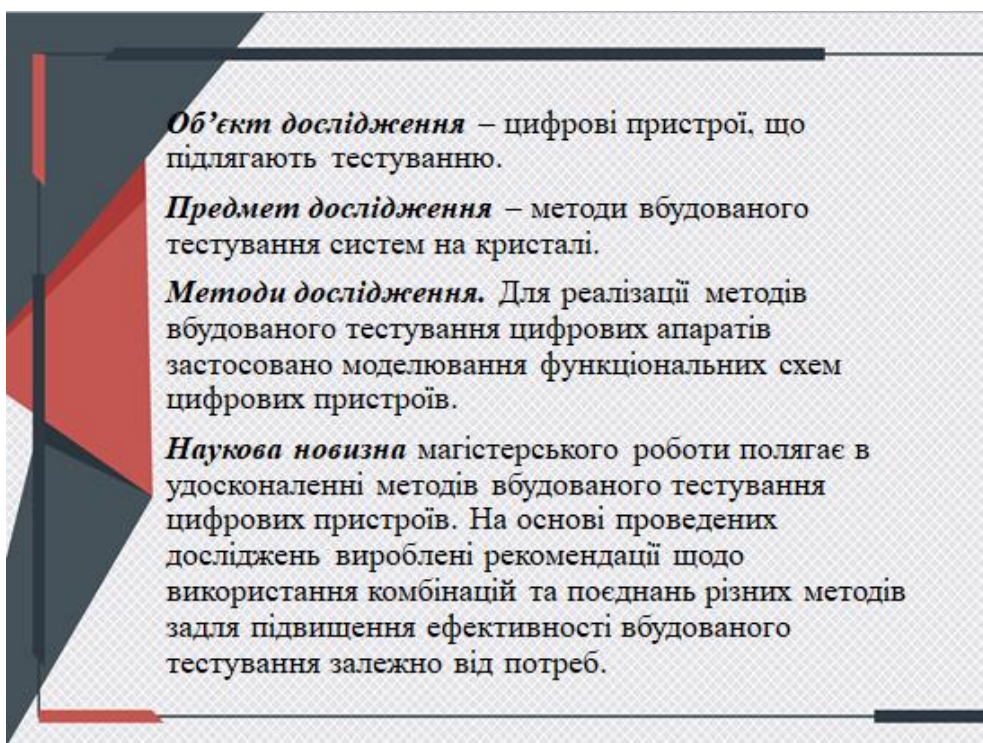


Рисунок Е.2 – Слайд 2

Основні задачі роботи

- Розглянути задачі діагностики цифрових схем.
- Провести аналіз існуючих методів реалізації вбудованого тестування.
- Визначити основні переваги та недоліки кожного із них.
- Оглянути основні моменти проектування схем на кристалі.
- Проаналізувати реалізацію та роботу методу граничного сканування.

Рисунок Е.3 – Слайд 3

Самоперевіряючі цифрові пристрої



Тут ДП – дискретний пристрій, СВК – схема вбудованого контролю. СВК аналізує входи X та виходи Y схеми та видає повідомлення про помилку на виході F .

Рисунок Е.4 – Слайд 4



Рисунок Е.5 – Слайд 5

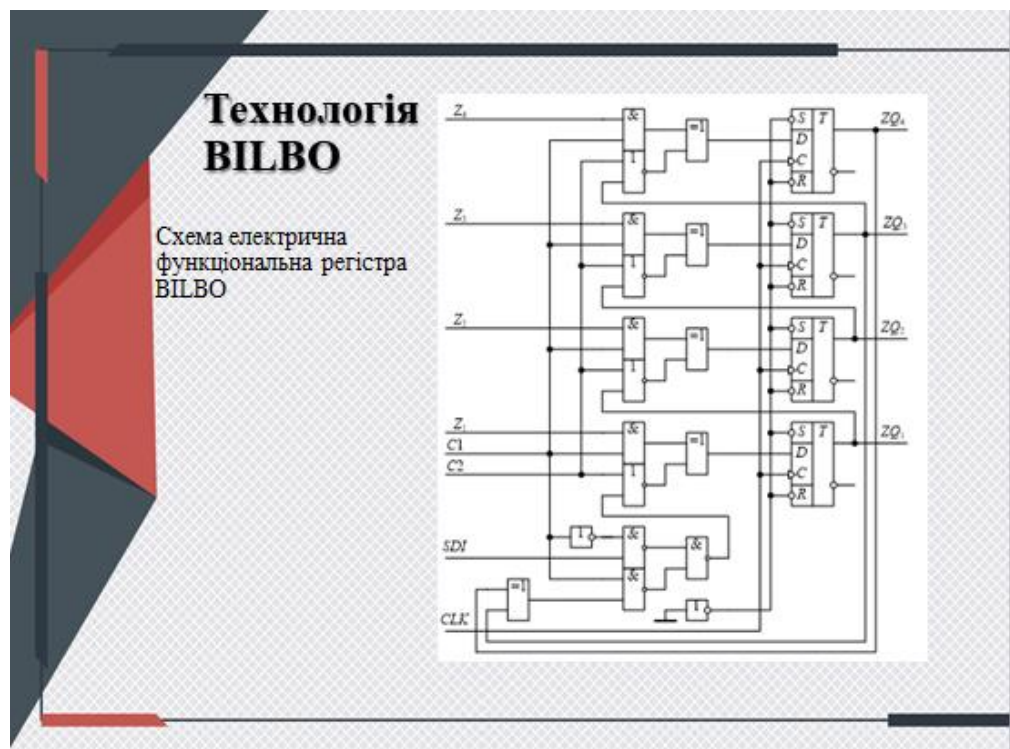


Рисунок Е.6 – Слайд 6



Рисунок Е.7 – Слайд 7



Рисунок Е.8 – Слайд 8



Рисунок Е.9 – Слайд 9



Рисунок Е.10 – Слайд 10

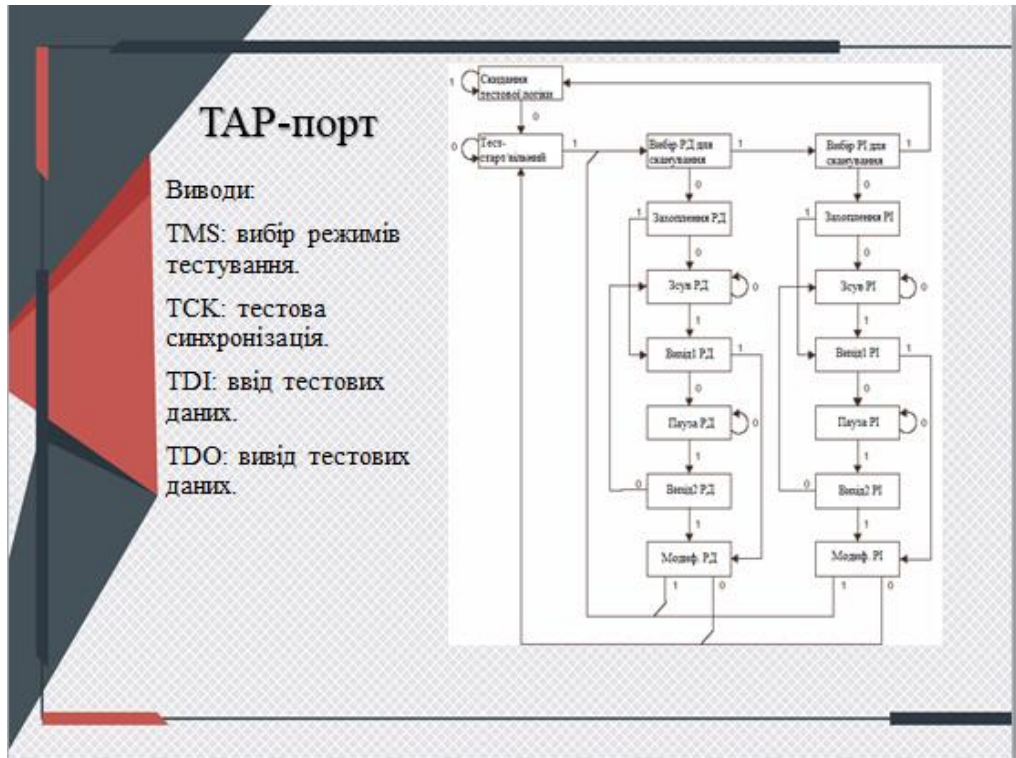


Рисунок Е.11 – Слайд 11



Рисунок Е.12 – Слайд 12



Рисунок Е.13 – Слайд 13



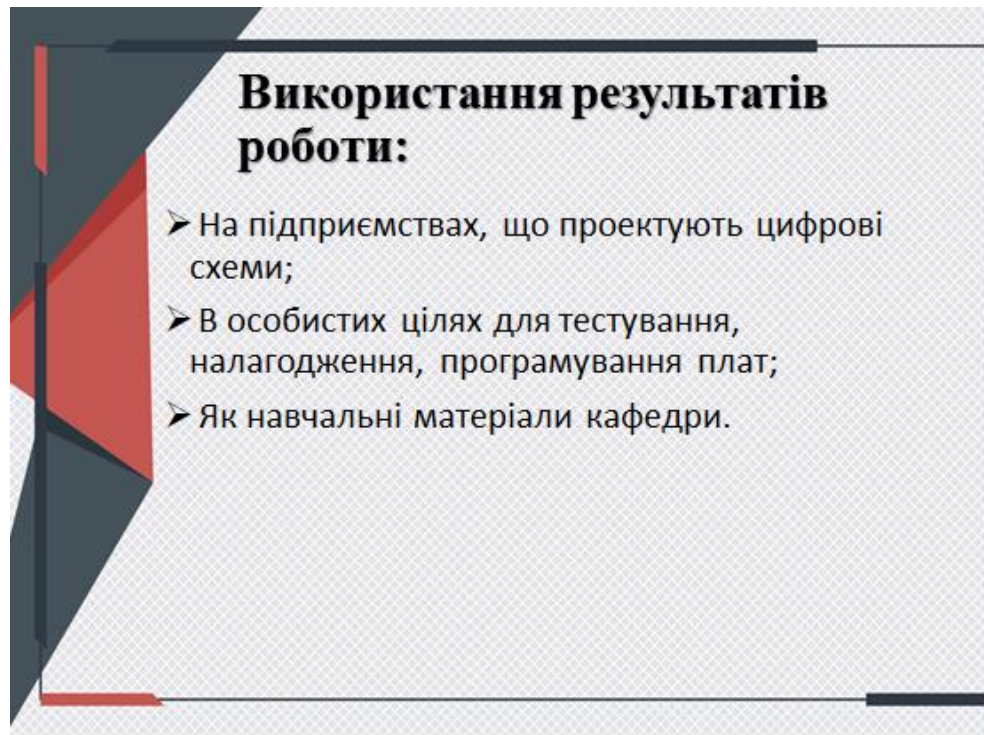
Рисунок Е.14 – Слайд 14



Рисунок Е.15 – Слайд 15



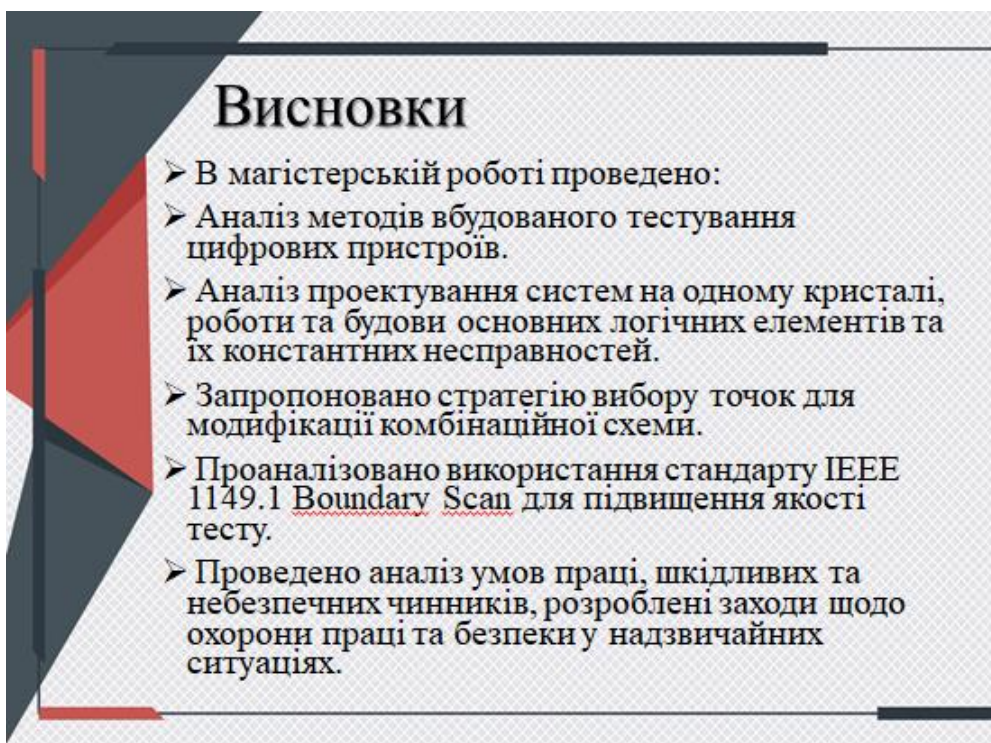
Рисунок Е.16 – Слайд 16



Використання результатів роботи:

- На підприємствах, що проектують цифрові схеми;
- В особистих цілях для тестування, налагодження, програмування плат;
- Як навчальні матеріали кафедри.

Рисунок Е.19 – Слайд 19



Висновки

- В магістерській роботі проведено:
- Аналіз методів вбудованого тестування цифрових пристроїв.
- Аналіз проектування систем на одному кристалі, роботи та будови основних логічних елементів та їх константних несправностей.
- Запропоновано стратегію вибору точок для модифікації комбінаційної схеми.
- Проаналізовано використання стандарту IEEE 1149.1 Boundary Scan для підвищення якості тесту.
- Проведено аналіз умов праці, шкідливих та небезпечних чинників, розроблені заходи щодо охорони праці та безпеки у надзвичайних ситуаціях.

Рисунок Е.20 – Слайд 20