Реферат

Мета роботи – розробити алгоритмічне та програмне забезпечення для тестування пакету кристалів ГАС, відповідно зі стандартом IEEE 1500.

Об’єкт роботи – пакет кристалів ГАС.

В кваліфікаційній роботі реалізовано алгоритмічне та програмне забезпечення для тестування пакету кристалів ГАС, відповідно зі стандартом IEEE 1500. Наведена інфраструктура відрізняється мінімальним набором процесів вбудованого діагностування в реальному масштабі часу та надає можливість здійснювати сервіси: тестування функціональностей на основі згенерованих вхідних послідовностей та аналізу вихідних реакцій; діагностування із заданою глибиною пошука дефектів у SoC; моделювання несправностей із метою виконання перших двох процедур на підставі таблиці несправностей.

Також розкрито структурно-алгебраїчний метод вбудованого діагностування дефектів у функціональних блоках SoC. Розглянутий метод використовує попередній аналіз таблиці несправностей із метою зменшення її об’єму та наступних обчислень, пов’язаних із побудовою ДНФ, яка формує усі рішення із встановлення діагнозу функціональностей SoC у реальному масштабі часу.

Система на кристалі, модуль пам’яті, діагностування дефектів, алгебро-логічний метод, моделювання несправностей, діагностика у реальному часі, функція мінімізації, тест-вектор, терм.

Abstract

The explanatory slip to degree work in volume 53 of sheets contains 10 figures and 25 of the references.

The purpose of a work – to develop algorithmic and software for testing a package of crystals flexible automated system in the correspondence with the standard IEEE 1500.

Plant of a work – package of crystals Flexible automated system.

In qualifying work is realized algorithmic and software for testing a package of crystals flexible automated system in the correspondence with the standard IEEE 1500. The reduced infrastructure differs by a minimum processes gang of firmware diagnosing in a real time scale and enables to realize services: testing functionals basis on generated entering sequences and analysis of output responses; diagnosing with the given depth of imperfections searching in SoC; modelling of inaccuracies with the purposes of first two procedures realization basis on trouble chart.

Also is uncovered the firmware diagnosing method of imperfections in function boxes SoC is structural-algebraic. The given method uses the preliminary analysis of the trouble chart with the purposes a diminution it volume and consequent evaluations connected to a construction disjunctive normal form which forming all solution on installation of functionals SoC-diagnosis in a real time scale.

System on crystal, module of memory, diagnosing of imperfections, algebraic-logic method, modelling of inaccuracies, diagnostics in real time, function of minimization, test-vector, term.

Список условных сокращений

АЛМ – алгебро-логический метод

АЛУ – арифметико-логическое устройство

ВЭП – вектора экспериментальной проверки

ГАС – гибкая автоматизированная система

ДНФ – дизъюнктивная нормальная форма

КНФ – конъюнктивная нормальная форма

ТН – таблица неисправностей

ATPG – Automated Test Pattern Generator

BIRA – Built-In Repair Analysis

BISR – Built-In Self Repair

BIST – Built-In Self Test

ESL – Electronic System Level

FDT – Fault Detection Table

F-IP – Functional Intellectual Propert

IEEE – Institute of Electrical and Electronics Engineers

I-IP – Infrastructure Intellectual Property

SoC – System-on-Chip

TLM – Transaction Level Modeling

Содержание

Введение

1. Анализ технического задания

1.1 Состояние рынка технологий сервисного обслуживания SoC

1.2 Структура сервисов SoC-микросхем

1.2.1 Модуль синтеза тестов

1.2.2 Модуль анализа неисправностей

2. Алгоритмическое и программное обеспечение тестирования пакета кристаллов ГАС

2.1 Алгебро-логический метод диагностирования неисправностей

2.2 Алгоритмизация АЛМ диагностирования неисправностей

2.3 Алгебро-логическая модель диагностирования F-IP

2.4 Уточнение диагноза F-IP, с помощью моделирования

2.5 Условное диагностирование F-IP на основе ДНФ

2.6 АЛМ для тестирования и ремонта SoC-памяти ГАС

2.7 Формализация АЛМ ремонта памяти

Выводы

Перечень ссылок

Введение

Вычислительная и аппаратная сложность современных гибких автоматизированных систем (ГАС), в основу организации которых заложены цифровые системы на кристаллах (System-on-Chip – SoC), характеризующиеся миллионами эквивалентных вентилей и требующих создания и внедрения новых высокоуровневых технологий проектирования – Electronic System Level (ESL) Design, моделинга – Transaction Level Modeling (TLM) и встроенного сервисного обслуживания – Infrastructure Intellectual Property (I-IP). Это означает, что поиск быстродействующих методов и средств приводит всех исследователей к необходимости повышения уровня абстракции моделей создаваемых функциональностей – Functional Intellectual Property (F-IP), встраиваемых в кристалл [1].

Рынок программных продуктов EDA уже предлагает инструменты для автоматизации процессов моделинга и верификации устройств системного уровня, начиная с компиляторов HDL-языков (C++, SystemC, SystemVerilog, UML, SDL) [2] и заканчивая графическими оболочками (Simulink, LabView, Xilinx EDK). Данные средства позволяют создавать проекты из существующих библиотечных компонентов путем использования ESL-мэппинга и создания TLM-интерфейсов [3, 4].

Рыночная привлекательность имплементации цифровой системы в кристалл FPGA определяется применением сравнительно дешевых чипов вместо универсальных процессоров, малой потребляемой мощностью, габаритными размерами, качественным и надежным выполнением основных функций, благодаря встроенной I-IP-инфраструктуре, что является актуальным в век мобильных вычислительных устройств.

Цель исследования – разработать алгоритмическое и программное обеспечение для тестирования пакета кристаллов ГАС, в соответствии со стандартом IEEE 1500.

Объект исследования – пакет кристаллов ГАС.

1. Анализ технического задания

1.1 Состояние рынка технологий сервисного обслуживания SoC

Проблема диагностирования и ремонта памяти связана с тенденцией на постоянное уменьшение площади кристалла, отводимой для оригинальной и стандартизованной логики с одновременным увеличением встроенной памяти. Как показано на рис. 1.1, увеличение удельного веса памяти на кристалле приводит к ее полному доминированию для хранения данных и программ, которое к 2014 году достигнет 94% [5]. Это обеспечит не только высокое быстродействие выполнения функциональности, но и гибкость, свойственную программному продукту в части коррекции ошибок проектирования.

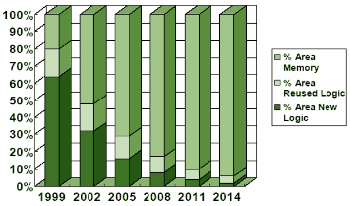


Рисунок 1.1 – Удельный вес SoC-памяти

Особенностью элементов памяти является тот факт, что в процессе их изготовления и эксплуатации отдельные ячейки под воздействием неисправностей могут выходить из состояния работоспособности. Данное обстоятельство не всегда приводит матрицу памяти к критическому состоянию, когда восстановление работоспособности невозможно. Поэтому далее рассматривается такое техническое состояние памяти, при котором суммарное количество дефектных ячеек не превышает резервных возможностей изделия, предназначенных для ремонта.

Современные технологии проектирования цифровых систем на кристаллах предлагают, наряду с созданием функциональных блоков F-IP, разработку сервисных модулей I-IP, ориентированных на комплексное решение проблемы качества проекта и повышение выхода годной продукции (Yield) в процессе производства, которое определяется внедрением в кристалл следующих сервисов [6]:

1) Наблюдение за состоянием внутренних и выходных линий в процессе функционирования, верификации и тестирования штатных блоков на основе использования стандарта граничного сканирования IEEE 1500 [7, 8];

2) Тестирование функциональных модулей путем подачи проверяющих наборов от различных тестовых генераторов, ориентированных на проверку дефектов или исправного поведения;

3) Диагностирование отказов и дефектов путем анализа информации, полученной на стадии тестирования и использования специальных методов встроенного поиска неисправностей на основе стандарта IEEE 1500;

4) Восстановление работоспособности функциональных модулей и памяти после фиксации отрицательного результата тестирования и определения места и вида дефекта при выполнении фазы диагностирования;

5) Измерение основных характеристик и параметров функционирования изделия на основе встроенных средств, позволяющих производить временные и вольтамперные измерения;

6) Надежность и отказоустойчивость функционирования изделия в процессе эксплуатации, которая достигается диверсификацией функциональных блоков, их дублированием и восстановлением работоспособности SoC в реальном масштабе времени.

1.2 Структура сервисов SoC-микросхем

На рис. 1.2 представлена усеченная структура [9, 10, 11], ориентированная на выполнение следующих задач:

1) Тестирование функциональностей на основе генерируемых входных последовательностей (Automated Test Pattern Generator – ATPG) и анализа выходных реакций;

2) Моделирование (Fault Simulator) неисправностей [12] в целях обеспечения диагностирования и ремонта на основе таблицы неисправностей (Fault Detection Table – FDT);

3) Диагностирование дефектов с заданной глубиной, путем использования мультизонда стандарта IEEE 1500;

4) Встроенный ремонт матричной памяти, на основе использования запасных компонентов (spare) [13].

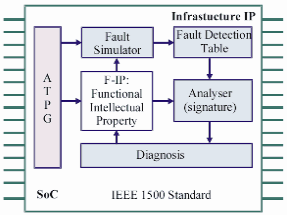


Рисунок 1.2 – Инфраструктура сервисов SoC DSP

1.2.1 Модуль синтеза тестов

Модуль синтеза тестов, предназначен для проверки функциональностей и одиночных неисправностей. В его состав входит набор генераторов входных последовательностей, обеспечивающих создание следующих тестов [14]:

PRTG – псевдослучайный генератор входных стимулов с равномерным законом распределения нулевых и единичных сигналов по входным переменным;

SATG – тестовый генератор шестнадцатеричных кодов на основе сигнатурного анализа;

SPTG – алгоритмический генератор тестовых векторов, активизирующих одномерные логические пути, ориентированные на проверку заданных одиночных неисправностей;

ADTG – тестовый генератор, предназначенный для проверки сумматорных схем АЛУ;

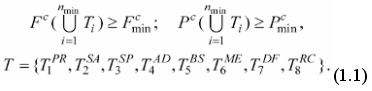
BSTG – тестовый генератор для шинных структур приема и передачи данных;

METG – генератор тестов, ориентированный на проверку матричной памяти;

DFTG – синтезатор тестов для автоматов, заданных в виде граф-схем алгоритмов;

RCTG – тестовый генератор для последовательностных счетно-регистровых структур и триггерных схем.

Модуль-генератор анализирует структурно-функциональную модель блока, подлежащего тестированию, и назначает подмножество таких синтезаторов, которые обеспечивают заданное качество покрытия дефектов (Fc) и функциональных режимов (Pc):



Обобщенная структура синтеза Testbench [14], представленная на рис. 1.3, включает также генератор HDL-кода, который предназначен для тестирования и верификации функциональностей на стадии разработки проекта.

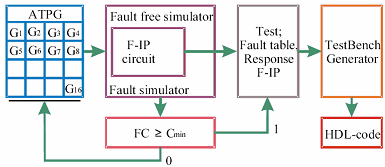


Рисунок 1.3 – Структура процесса синтеза Testbench для F-IP

Количество тестовых генераторов на стадии проектирования SoC может быть существенно большим, чем то подмножество, которое далее встраивается в кристалл. Поэтому на стадии моделирования и верификации выполняется анализ покрывающих свойств каждого тест-генератора в целях поиска их минимальной совокупной конфигурации, которая удовлетворяет выражению (1.1).

Важно отметить, что в ближайшие 5 лет идеология синтеза тестов для цифровых систем на кристаллах будет заимствовать лучшие традиции, идущие от ESL- и TLM-проектирования [15]:

1) Использование библиотек тестов (Testbench) ведущих компаний для тестирования и верификации стандартизованных функциональностей, обозначенных в качестве F-IP;

2) Применение стандартных решений сервисного обслуживания I-IP для встроенного тестирования компонентов системы на кристалле;

3) Создание собственных библиотек тестов для вновь разрабатываемых функциональностей;

4) Внедрение новой технологии синтеза тестов для цифровой системы, основанной на дискретном мэппинге покрытия функциональностей и дефектов исходной спецификации с помощью минимальной совокупности Testbench, из библиотеки тестов (рис. 1.4);

5) Применение встроенных средств тестопригодности, таких как IEEE boundary scan – средства граничного сканирования, и шести компонентов I-IP для повышения технологичности процедур синтеза тестов.

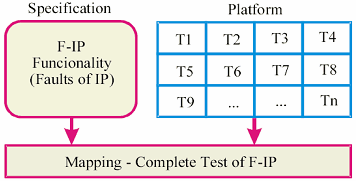


Рисунок 1.4 – Mapping модели синтеза тестов для F-IP

1.2.2 Модуль анализа неисправностей

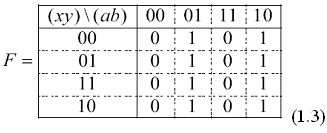
Модуль анализа неисправностей использует дедуктивный алгоритм, ориентированный на проверку одиночных дефектов, генерируемых на основе аналитического или табличного описаний функциональностей SoC. Это означает, что дедуктивное моделирование может обрабатывать проекты, представленные как на вентильном, так и на любом другом, более высоком уровне абстракции (регистровом, системном).

Основная идея метода заключается в создании дедуктивной модели функциональности на основе использования известного выражения [12]:

(1.2)



где дедуктивная функция F на тест-векторе T есть модифицированное описание исправного поведения, позволяющее вычислять списки входных неисправностей, транспортируемые на выход схемы под воздействием входных сигналов. На примере функции Xor демонстрируется синтез дедуктивной функции по карте Карно:



Переменные xy – булевы, а сигналы ab – (регистровые) для записи списков дефектов:

(1.4)



Аппаратная реализация дедуктивной функции, представленной формулой (1.4), изображена на рис. 1.5.

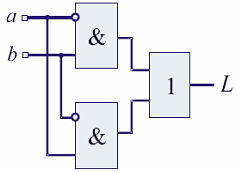


Рисунок 1.5 – Дедуктивный примитив функции Xor

Схемный примитив является универсальным по отношению к различным тестовым последовательностям. Стратегия, рассмотренная в данной квалификационной работе относительно синтеза моделей, основывается на создании библиотеки дедуктивных элементов, покрывающих все стандартизованные конструктивы функциональностей, которыми оперирует разработчик, создавая в автоматизированном режиме проект в виде цифровой системы на кристалле. В данном случае речь идет о синтезе дедуктивной структуры на основе мэппинга, суть которого представлена на рис. 1.6.

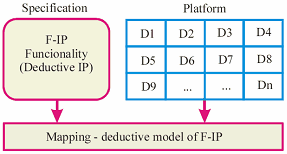


Рисунок 1.6 – Mapping дедуктивной модели для F-IP

Предложенный в [6] подход к дедуктивному анализу, предполагает создание на кристалле еще одной встроенной модели, которая должна обеспечивать практически все шесть сервисов, предусмотренных стандартом инфраструктуры I-IP.

Платой за качество диагностического и тестового обслуживания является достаточно высокая стоимость дополнительных аппаратурных затрат, которые превышают штатную функциональность в 10 – 15 раз. При этом выигрыш в быстродействии, по сравнению с внешней программной реализацией дедуктивного анализа, составляет 2 – 3 порядка, что практически обеспечивает сервисное обслуживание в реальном масштабе времени.

Другое, более экономичное решение проблемы, связано с интерактивной модификацией схемной структуры дедуктивной модели для каждого тест-вектора. Для этого используется внутренняя память кристалла, где формируется модель по правилам, определенным в (1.2). Мэппинг (см. рис. 1.6) дает дедуктивную функцию, где аппаратные затраты равны стоимости функциональности F-IP.

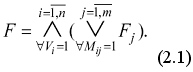
2. Алгоритмическое и программное обеспечение тестирования пакета кристаллов ГАС

2.1 Алгебро-логический метод диагностирования неисправностей

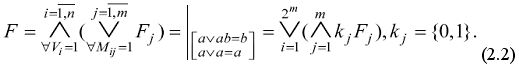
В данном методе, основная роль отводится технологии граничного сканирования, которая, в настоящее время, внедренная в кристалл, призвана облегчить решение практически всех задач сервисного обслуживания функциональных модулей системы на кристалле.

Контроллер доступа к внутренним линиям и портам регистра граничного сканирования использует ячейку или разряд регистра. В совокупности, число таких ячеек, обеспечивающих в данном случае мониторинг, должно быть равно количеству проблемных наблюдаемых линий проекта, которые необходимы для точного установления диагноза.

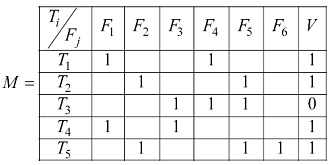
Основанная на регистре граничного сканирования процедура диагностирования использует также информацию из таблицы неисправностей (ТН), которая представляет собой множество дефектов, покрываемых тестовыми наборами. Используя информацию о результате проведения диагностического эксперимента, которая представлена в виде вектора экспериментальной проверки (ВЭП): а также таблицу [12] неисправностей F, выполняется процедура установления диагноза по выражению, записанному в форме произведения дизъюнкций всех дефектов [16], которые могут дать экспериментальную реакцию в виде V, определенном единичными и нулевыми значениями:



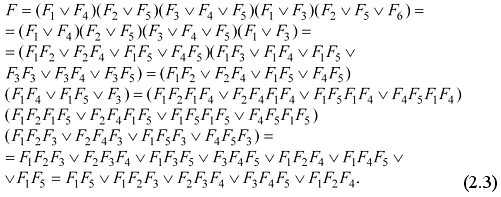
Полученная из таблицы неисправностей КНФ трансформируется к ДНФ с помощью эквивалентных преобразований (логическое умножение, минимизация и поглощение) [11, 16]. В результате получается булева функция, где термы – логические произведения – представляют полное множество решений в виде сочетания дефектов (дающих по выходам SoC или ее компоненту вектор экспериментальной проверки V):



Представленная процедура, в общем случае, диагностирует некоторое подмножество дефектов, которое в дальнейшем нуждается в уточнении путем применения дополнительного зондирования внутренних точек с помощью регистра граничного сканирования. Пример поиска дефектов рассматривается на основе следующей ниже таблицы неисправностей (столбцы – дефекты, строки – тестовые последовательности), которая является продуктом дедуктивного анализа дефектов и вектора экспериментальной проверки [17]:



Количество единиц в ВЭП V, формирует число дизъюнктивных термов КНФ (2.2). Каждый терм – построчная запись дефектов (через логическую операцию ИЛИ), оказывающих влияние на выходы функциональности. Уже само представление таблицы в виде аналитической записи – КНФ – дает потенциальную возможность существенно сократить объем диагностической информации для поиска дефектов. Тем более, последующее преобразование КНФ к ДНФ на основе тождеств алгебры логики позволяет существенно уменьшить булеву функцию, что иллюстрируется следующим результатом:



Для уменьшения количества вычислений при выполнении логического умножения в первой строке (2.3) исходную запись можно упростить согласно законам булевой алгебры:

(2.4)



Полученный результат предоставляет все возможные решения – покрытия дефектами строк таблицы неисправностей в функциональности SoC, при условии, что ВЭП имеет все единичные координаты V = (11111). Принимая во внимание фактическое значение ВЭП, равное V = (11011), выполняется моделирование функции F путем подстановки нулевых значений дефектов, которые теоретически проверяются, но дают в векторе V нулевую координату. Такими являются дефекты:

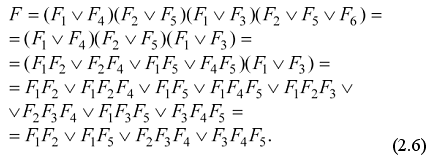


Окончательный результат определяется следующей функцией:

(2.5)



Любое сочетание – конъюнктивный терм ДНФ, приведенный в решении , покрывает все строки таблицы неисправностей по определению, поэтому введение любой нулевой строки обязательно обращает в ноль функцию F. Поэтому корректное решение, соответствующее вектору экспериментальной проверки, должно изначально учитывать нулевые координаты вектора V. С учетом сказанного из выражения (2.3) на стадии записи КНФ необходимо исключить терм



Результат представляет все возможные решения, которые приводят к реакции изделия, определенного заданным ВЭП:

(2.7)



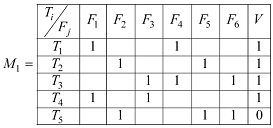
Дополнительное моделирование последней булевой функции дает окончательное решение в виде сочетания двух дефектов:

(2.8)



2.2 Алгоритмизация АЛМ диагностирования неисправностей

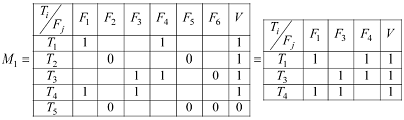
АЛМ может быть формально рассмотрен на примере следующей таблицы неисправностей M1 и представлен в виде пяти пунктов алгоритма:



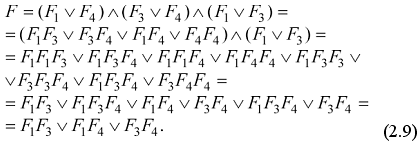
1. Определение всех строк, соответствующих нулевым значениям ВЭП в целях обнуления всех единичных координат найденных строк. В данном случае – это одна строка T5.

2. Нахождение всех столбцов, которые имеют нулевые значения координат строк с нулевым состоянием ВЭП. Обнуление единичных значений найденных столбцов. В данном случае: F2, F5, F6.

3. Удаление из таблицы неисправностей строк и столбцов, имеющих только нулевые значения координат (найденные в пунктах 1 и 2).



4. Построение КНФ по единичным значениям ВЭП:



5. Преобразование КНФ к ДНФ с последующей минимизацией функции. В данном случае это приводит к получению искомого результата в виде сочетания неисправностей:

(2.10)



Предложенный алгоритм ориентирован на предварительный анализ ТН, в целях уменьшения объема и последующих вычислений, связанных с построением ДНФ, формирующей все решения по установлению диагноза функциональностей SoC. Дальнейшее уточнение диагноза возможно только с применением мультизонда на основе регистра граничного сканирования данных [7].

2.3 Алгебро-логическая модель диагностирования F-IP

Структура модулей сервисного обслуживания I-IP для диагностирования дефектов в функциональных блоках F-IP представлена на рис. 2.1. Компаратор () анализирует выходные реакции модели и реального устройства на входные тестовые векторы, поступающие от генератора тестов. Несовпадения модельных и экспериментальных реакций на тесте формируют единичные координаты ВЭП для каждого входного набора. Взаимодействие ВЭП с ТН ( размерностью число тест-векторов, n – количество разрядов boundary scan регистра) и схемной структурой дают множество линий и элементов, подозреваемые как дефектные на текущем тест-векторе.

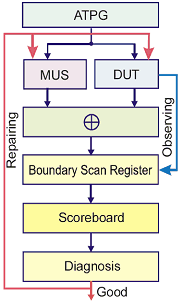
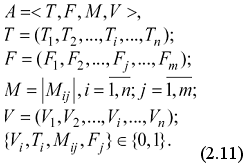


Рисунок 2.1 – Модель процесса диагностирования F-IP

Для организации вычислительных процессов, приводящих к точному диагнозу, чрезвычайно важна метрика или форма представления исходной информации.

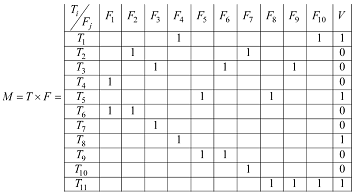
Интересное решение задачи диагностирования может быть получено путем применения булевой алгебры и таблицы неисправностей M, представляющей собой декартово произведение теста Т на множество заданных дефектов F, в совокупности с ВЭП V, где выполнение задачи покрытия дает максимально точный результат в виде ДНФ, а каждый терм есть возможный вариант наличия в устройстве дефектов. Итак, модель процесса диагностирования представлена компонентами:



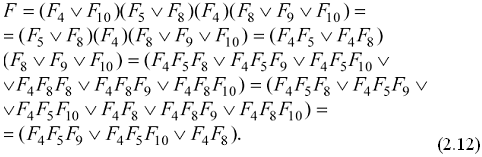
Решение задачи диагностирования сводится к анализу ТН, полученной в результате моделирования дефектов, путем записи логического произведения дизъюнкций (КНФ), записанных по единичным значениям строк таблицы неисправностей (2.1). Далее КНФ трансформируется к ДНФ (2.2) с помощью эквивалентных преобразований. В результате получается булева функция, где термы – логические произведения – есть полное множество решений, представляющее собой сочетания дефектов, дающие по выходам функциональности ВЭП, полученный в результате выполнения диагностического эксперимента.

Следующая матрица M = T × F является примером алгебро-логического анализа дефектов на основе ТН в функциональных блоках системы на кристалле, число которых равно 10. Тест, длиной 11 входных наборов, проверяет все введенные в таблицу неисправности. Вектор экспериментальной проверки цифрового устройства V = (10001001001), полученный при выполнении диагностического эксперимента, фиксирует несовпадения выходов устройства по сравнению с моделью (золотым эталоном) на четырех (1, 5, 8 и 11) тестовых наборах.

В соответствии с числом единиц в ВЭП V, формируется количество дизъюнктивных термов КНФ, равное 4. Каждый терм есть построчная запись дефектов через логическую операцию ИЛИ, оказывающих влияние на искажение выходных сигналов функциональности.



Далее осуществляется преобразование КНФ к ДНФ на основе правил алгебры логики, что дает возможность получить результат:



Полученный результат

(2.12)

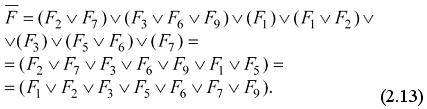


содержит во всех термах дефект F4, означающий его обязательное присутствие в функциональности SoC. Если принять гипотезу о существовании одиночного или минимального числа кратных дефектов, то предпочтительным является решение, определяемое третьим термом – в схеме существует два дефекта, которые формируют на выходах ВЭП, равный V = (10001001001).



2.4 Уточнение диагноза F-IP, с помощью моделирования

Полученная дизъюнктивная форма (2.2) является основной моделью для поиска дефектов. Она не всегда однозначно определяет дефект функциональности, поэтому нуждается в процедурах, уточняющих диагноз. Прежде всего, следует заметить, что все строки M = T × F, которые отмечены нулевыми значениями ВЭП, можно объединить в дизъюнкцию неисправностей (2.2). Получение формы (2.1) из рассматриваемой ТН дает возможность определить все дефекты, которые не могут присутствовать в схеме:



Анализ выражений, представленных формулами (2.12) и (2.13) приводит к следующим выводам:

1) Дефекты, которые не могут присутствовать в схеме, определяются в термах ДНФ, полученных по нулевым строкам относительно ВЭП;

2) Дефекты, которые имеются в ДНФ, должны быть удалены из функции (2.13);

3) Исключение в данном случае дефекта F5 приводит к разрушению двух термов поскольку без неисправности F5 каждый из них, в отдельности, не сможет сформировать заданный ВЭД;



4) Таким образом, делается единственный вывод – в схеме присутствует двукратная ошибка, определяемая термом ;



5) Вычислительная сложность получения точного и полного множества решений определяется выражением – число дефектов.



Обозначив отсутствие конкретной неисправности Fi = 0, можно сформировать входные условия для ДНФ (2.12) в целях последующей эмуляции (моделирования) функции при следующих начальных условиях:

(2.14)



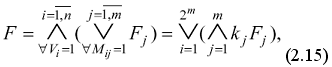
Тогда результат моделирования функции становится равным .



В самом деле, если неисправности теоретически проверяемые на тестовых наборах, дают отрицательный результат – все они не искажают состояния выходов, то значит – они отсутствуют в схеме. Обоснование данного факта подтверждается следующими доказательствами.



Лемма 1. Полное множество всех возможных сочетаний дефектов, проверяемых тестом T, определяется как ДНФ, полученная преобразованием конъюнктивной формы:



каждый терм которой записан по единичным значениям строки ТН [17] M = T × F, имеющей состояние ВЭП Vi = 1.

Исходная информация, записанная в соответствии с единичными значениями ВЭП, представляет собой полную модель неисправного поведения реального объекта, которая формирует ВЭП с фиксированным числом единиц (строк ТН), равным k. Каждая строка формирует дизъюнкцию дефектов, записанную по ИЛИ. Число таких дизъюнкций равно k, которые логически перемножаются, образуя полное и непротиворечивое множество событий – дефектов, одновременно присутствующих в схеме. Путем перемножения элементарных дизъюнкций с последующим упрощением выражений, используя аксиомы , получается ДНФ, которая создает все возможные сочетания, записанные в виде элементарных конъюнкций. Ввиду тождественности выполненных преобразований полученная функция по логике эквивалентна исходной КНФ, а, по сути, есть технологичная форма записи всех решений – сочетаний дефектов, имеющих место быть в схеме.



Лемма 2. Все проверяемые в строках таблицы неисправностей M = T × F дефекты, отмеченные нулевыми значениями ВЭП Vi = 0, в реальном объекте отсутствуют.

В самом деле, ТН M = T × F имеет два типа строк: единичные и нулевые относительно значения ВЭП:

(2.16)



Строка p выявляет наличие в схеме двух дефектов . Строка q свидетельствует о теоретической проверке неисправностей , если бы вектор был равен 1: Vq = 1. Но фактически сигнал Vq = 0 идентифицирует несущественность дефектов для искажения выходов схемы или данные дефекты отсутствуют в тестируемом изделии. Подставив в функцию нулевые сигналы для , получаем результат: .



Аналогично, все дефекты, которые определены в строках, имеющих нулевое значение ВЭП, будут отсутствовать в схеме. Но если это так, то их следует исключить из ДНФ, записанной по единичным значениям ВЭП. Следовательно, имея термы ДНФ и множество дефектов, которые не могут существовать в схеме для заданного ВЭП, можно выполнить процедуру подстановки нулевых сигналов в переменные элементарных конъюнкций функции ДНФ. Но с учетом того факта, что, , результат подстановки и последующих преобразований, в целях получения минимальной функции, будет иметь только те термы, которые не имеют переменных – дефектов с нулевым значением сигналов. Это означает, что из ДНФ будут исключены все дефекты, относящиеся к нулевым, относительно ВЭП, строкам таблицы неисправностей.

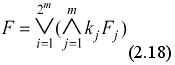


Теорема 1. Минимальное множество всех возможных сочетаний дефектов, определяемых по ТН M = T × F, вычисляется путем моделирования ДНФ на множестве начальных условий:



определенных нулевыми значениями всех проверяемых дефектов, соответствующих нулевым сигналам ВЭП.

В соответствии с леммой 1 полное множество всех возможных сочетаний дефектов, проверяемых тестом, определяется в виде ДНФ:



которая формирует все решения, удовлетворяющие единичным значениям вектора экспериментальной проверки Vq = 1. Оно может быть уменьшено за счет исключения тех дефектов, которые теоретически проверяются тестом, но фактически, на тестовых наборах они не искажают состояний выходов, что означает их отсутствие в реальной схеме. Следовательно, их можно исключить из термов ДНФ, образующих полное множество всех возможных сочетаний. Механизм такого исключения, согласно лемме 2, заключается в подстановке нулевых значений переменных в термы ДНФ с последующим моделированием (упрощением) функции. Если терм имеет 0-компонент некоторой переменной Fi, то, согласно алгебре логики, весь терм обращается в 0, что означает его исключение из ДНФ.

В результате минимизации, на основе учета условий леммы 2, остается минимальная ДНФ, содержащая наименьшее число возможных сочетаний дефектов (одиночных и кратных), которое невозможно сократить без использования дополнительной диагностической информации, поступающей от мультизонда на основе boundary scan-регистра.

Таким образом, предложенный АЛМ диагностирования использует булево исчисление в качестве базового аппарата для решения задачи покрытия путем получения дизъюнктивной формы, минимизирующейся путем исключения термов, имеющих переменные дефектов, относящиеся к строкам с нулевыми значениями ВЭП. Для незначительного числа дефектов в цифровой системе на кристалле вычислительная сложность позволяет осуществлять поиск неисправностей в реальном масштабе времени.

2.5 Условное диагностирование F-IP на основе ДНФ

В целях существенного уменьшения области подозреваемых дефектов, на практике используется метод половинного деления [16], основанный на использовании интерактивной процедуры зондирования внутренних точек контроля, которые обеспечивают полученную ДНФ дефектов дополнительной информацией для уменьшения множества дефектов. В данном случае таким зондом может быть регистр граничного сканирования, который способен определить состояние внутренней линии в целях исключения дефектов или их подтверждения.

Стратегия выбора контрольной точки ориентирована на примерно половинное деление подозреваемого множества – исключение на каждом шаге половины дефектов путем моделирования – упрощения исходной ДНФ. Суть метода половинного деления на ДНФ, представляющей все возможные сочетания дефектов в схеме, можно продемонстрировать на следующем примере.

Выбор первой точки контроля, например F9 = 0, превращает булеву функцию в уменьшенное выражение:



(2.19)



Если же F9 = 1, что означает подтверждение дефекта на линии, то уменьшения размерности ДНФ не происходит. Необходимо ориентировать алгоритм выбора точек контроля на максимальное уменьшение исходной ДНФ после введения начальных условий (Fj = {0, 1}) для моделирования. Критерием выбора точки контроля может служить взвешенность мощностей ДНФ, полученных после моделирования обоих состояний проверки.

Правила выбора контрольной точки регламентируются следующими утверждениями.

Утверждение 1. Если Fj присутствует во всех термах ДНФ, то данный дефект существует обязательно в схеме – его не следует тестировать. В противном случае, если предположить, что проверка будет равна нулю, то все термы обращаются в нуль, а это противоречит условию существования ненулевых значений ВЭП V.

Утверждение 2. В схеме присутствует только одно сочетание дефектов, определенное одним термом ДНФ. Если найдено одно подтвержденное решение в виде терма ДНФ, то остальные термы следует исключить из рассмотрения путем их обращения в нуль.

Поэтому задача минимизации точек контроля сводится к выполнению двух альтернативных стратегий:

1) Рассмотрение переменных в термах минимальной длины для подтверждения всех дефектов в терме путем их зондирования;

2) Проверка таких переменных, которые обращают в нуль максимальное число термов ДНФ.

В случае существования функции , которая имеет терм минимальной длины 2, а также переменную F4 во всех термах, единственно лучшим решением будет проверка F8, которая дает при положительном результате искомое множество дефектов, а при отрицательном – оставшиеся два терма, подлежащие зондированию:



(2.20)



Проверка F5 дает следующие результаты послезондового моделирования двух вариантов функций:

(2.21)



Далее, после (F5 = 1), должны последовать две проверки из трех (F9, F10, F8), которые убирают все термы, кроме одного, определяющего решение:

(2.22)



Критерием окончания процедуры диагностирования является получение одного терма ДНФ, которое идентифицирует наличие кратного дефекта в функциональности цифровой системы на кристалле.

Далее предлагается еще один пример выполнения интерактивной процедуры диагностирования на основе анализа ДНФ:

(2.23)



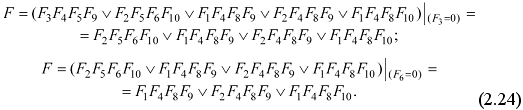
В устройстве существует кратная неисправность



Выполняется подсчет весов каждой переменной, входящей в ДНФ:



2) Вероятность присутствия в схеме дефектов коррелируется с их весовыми коэффициентами. Следовательно, в целях получения единственного решения в виде терма ДНФ, необходимо выбирать, в качестве точек контроля, переменные, имеющие минимальный вес, обращающие термы в нулевые составляющие. Следуя сказанному, первая и вторая точки контроля есть (F3, F6), имеющие минимальный вес:



3) После каждого шага выполняется перерасчет весовых коэффициентов, корректирующий последующие шаги:



Здесь установлен факт наличия в схеме дефектов (F4, F8), которые уже не подлежат зондированию в соответствии с условием утверждения 1.

Проверка дефекта F2 дает следующий результат:

(2.25)



Пересчет коэффициентов:



предполагает наличие в схеме дефектов (F1, F4, F8) и дополнительную проверку одной из линий (F9, F10):

(2.26)



Таким образом, в результате выполнения четырех зондирований, представленных линиями (F3, F6, F2, F9), был получен точный диагноз – в схеме присутствует кратный дефект:



2.6 АЛМ для тестирования и ремонта SoC-памяти ГАС

В процессе производства и эксплуатации любых видов памяти, используемой в ГАС, необходимы гарантии ее соответствия техническим условиям. Для этого предусмотрено выполнение трех процедур:

1) Тестирование памяти, заключающееся в подаче тестовых воздействий, ориентированных на выявление определенных классов дефектов [5, 6];

2) В случае возникновения неисправности, необходима дополнительная процедура диагностирования, позволяющая определить место, причину и вид дефекта;

3) После определения множества дефектов, препятствующих выполнению функции памяти, необходимо активизировать процесс восстановления работоспособности – замену дефектных элементов избыточными резервными компонентами, изначально находящимися на силиконовом кристалле [9, 13].

Отсюда следует, что упомянутые действия ориентированы на повышение выхода годных изделий (Yield) без существенных дополнительных временных и материальных затрат. Для восстановления работоспособности необходим специальный механизм ремонта памяти путем замены дефектных компонентов на исправные из резерва силиконового кристалла.

Процедура тестирования, как правило, осуществляется с помощью BIST-блока (Built-In Self Test), который представляет аппаратный быстродействующий генератор тестовых наборов, а также анализатор (сигнатурный) реакций выходов памяти на тестовые последовательности. Анализ восстановления (Repair Analysis) заключается в определении возможности покрытия дефектных элементов памяти, существующими в наличии резервными компонентами. Модуль памяти представлен двумя частями:

1) Функциональные ячейки, которые непосредственно применяются для хранения данных и программ при использовании модуля в системе на кристалле;

2) Резервные или запасные ячейки, которые предназначены для восстановления работоспособности памяти в случае отказов функциональных ячеек.

Функциональные и резервные ячейки объединяются в столбцы и строки. При обнаружении дефекта строка (столбец), содержащая дефектный элемент, отключается от функциональной структуры ячеек памяти, а на ее место подключается строка (столбец) из резерва кристалла. Поскольку количество резервных компонентов ограничено, необходим специальный механизм, позволяющий эффективно распределять ресурсы восстановления работоспособности в целях обеспечения покрытия дефектных элементов памяти минимально возможным количеством избыточных столбцов и строк.

Описанная выше процедура поиска, может быть реализована как в качестве встроенного модуля восстановления работоспособности, так и внешнего – по отношению к кристаллу. Во втором случае, данные об ошибках поступают извне, обрабатываются и передаются контроллеру, обеспечивающему восстановление памяти. Это приводит к значительным потерям времени. Поэтому предпочтение отдается on-chip реализации модуля, когда данные об ошибках передаются непосредственно из BIST. Такой механизм носит название BIRA (Built-In Repair Analysis) [9, 11] – встроенный анализ восстановления работоспособности.

Ремонт памяти осуществляется с помощью отключения дефектных элементов (строк и столбцов матрицы) путем электрического плавления перемычек и подключения резервных. Процесс пайки может быть электрическим или лазерным [12]. Устройство электрической пайки имеет меньшие габаритные размеры, чем лазерной, и применяется чаще. Пайка перемычек выполняется с помощью набора инструкций, хранящихся в постоянной памяти внутри чипа (hard repair) или в оперативной памяти (soft repair) [9, 10, 13].

Soft repair имеет ряд преимуществ: при возникновении ошибки новая исправленная инструкция может быть легко записана в память; обеспечивается экономное использование площади кристалла и достаточная надежность [19].

Hard repair позволяет использовать упрощенный производственный тест и обеспечивает обнаружение ошибок, которые в силу определенных обстоятельств не могут быть зафиксированы при soft repair, например перегрев.

Структура процессов встроенного анализа и (soft repair) самовосстановления памяти – BISR (Built-In Self Repair) – [9 – 10] представлена на рис. 2.2.

1) Активизация чипа, заполнение регистра BISR нулевыми значениями.

2) Запуск контроллера BIST. Тестирование памяти и накопление информации о дефектных ячейках в регистре BIRA.

3) Передача информации о дефектных ячейках в регистр BISR для последующей перепайки.

4) Сканирование BIRA-регистров, содержащих статус восстановления, контроллером BIST для получения информации о дефектах.

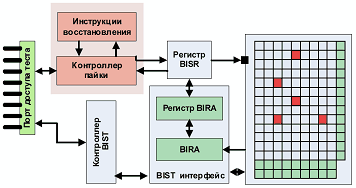


Рисунок 2.2 – Схема встроенного анализа и восстановления памяти

5) Запуск контроллера пайки в режиме записи и передача из BISR инструкций восстановления.

6) Перезагрузка чипа. Запись в регистр BISR информации о пайке перемычек, замена дефектных строк и столбцов резервными компонентами.

7) Запуск контроллера BIST в целях повторного тестирования памяти и проверки правильности результата восстановления.

Функция цели Z данного исследования, исходя из современных достижений в области оперативного восстановления памяти, может быть сформулирована следующим образом: минимизация стоимости восстановления (аппаратурных затрат) модуля памяти M = | Mij | в процессе эксплуатации систем на кристаллах путем использования АЛМ минимизации покрытия множества дефектных ячеек памяти системой резервных элементов в условиях ограничений N на количество последних:

(2.27)



где – стоимость i-го варианта решения восстановления модуля памяти M = | Mij | с помощью минимального подмножества строк и столбцов резерва кристалла, покрывающего множество F дефектных ячеек памяти



Далее рассматривается метод получения минимального покрытия на примере матрицы памяти с пятью дефектными ячейками [11], двумя резервными строками и одним столбцом (см. рис. 2.3).

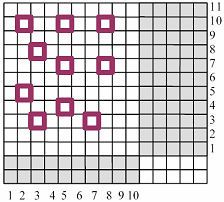


Рисунок 2.3 – Матрица памяти с дефектными ячейками и резервом

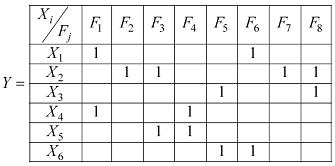
Каждый резервный компонент (строка или столбец) способен восстановить работоспособность от одной до n дефектных ячеек, принадлежащих строке или столбцу.

Идея метода сводится к оптимальному замещению дефектных элементов матрицы памяти, путем решения задачи покрытия дефектов-столбцов резервом строк. Для иллюстрации метода первоначально предлагается воспользоваться матрицей покрытия заданных неисправностей F некоторым количество строк (это могут быть тестовые наборы, резервные строки) X, причем:

(2.28)



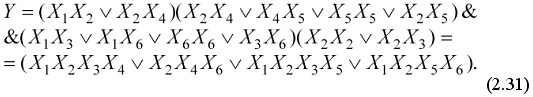
Пусть задана матрица Y, имеющая вид :



Точное решение задачи покрытия неисправностей минимальным числом резервных строк памяти основывается на синтезе булевой функции, записываемой как конъюнкция дизъюнкций по конституентам единиц, соответствующих столбцам приведенной выше матрицы:



В данном случае аналитическая запись в виде булевой функции, представленной в виде КНФ, есть исходная модель, содержащая полное множество решений задачи покрытия, которая решается путем нахождения ДНФ. Для этого выполняется процедура преобразования КНФ в ДНФ путем перемножения всех термов. В результате эквивалентных преобразований, выполненных по правилам алгебры логики, получается булева функция, содержащая все возможные покрытия неисправностей, описанные с помощью четырех вариантов комбинаций строк:



Минимальное решение задачи покрытия содержит всего три резервных строки, с помощью которых покрывается 8 дефектов в матрице памяти:



Для использования предложенного метода восстановления работоспособности памяти, необходимо иметь в виду, что каждый дефект Fi в матрице памяти принадлежит как строке, так и столбцу. Поэтому преобразование топологической модели дефектов памяти к матрице покрытия неисправностей, заключается в присвоении каждому дефекту номеров строк и столбцов, которые искажаются данной неисправностью.

Для примера (рис. 2.4), где имеется 5 дефектных ячеек, покрываемых тремя столбцами и четверкой строк.

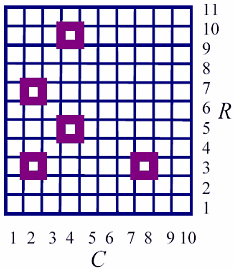
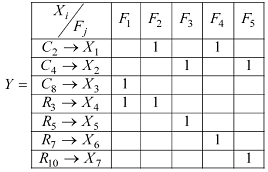


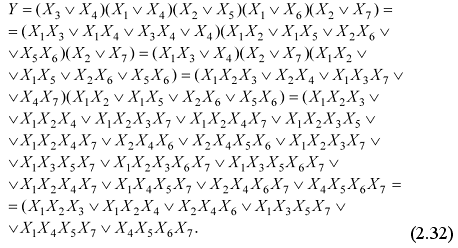
Рисунок 2.4 – Матрица памяти с дефектными ячейками

Преобразование трансформирует матрицу памяти к таблице покрытия, где левый столбец задает взаимно-однозначное соответствие между координатами дефекта, в номерах строк и столбцов матрицы памяти и строками покрытия неисправностей:



Иначе, топология матрицы памяти из двумерной метрики трансформируется в одномерную структуру строк, обладающими определенными покрывающими свойствами, относительно столбцов неисправностей.

Последующая запись булевой функции формирует логическое произведение дизъюнкций, записанных по конституентам единиц, соответствующих столбцам упомянутой выше матрицы :

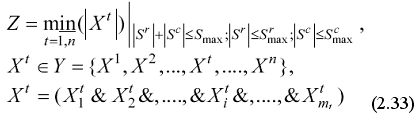


Эквивалентные преобразования позволили упростить достаточно сложную конструкцию – КНФ – с получением минимального множества всех решений, число которых, в данном случае, равно шести. Подмножество минимальных решений определяется тремя конъюнктивными термами, каждый из которых содержит 3 резервных элемента для восстановления работоспособности матрицы памяти:



2.7 Формализация АЛМ ремонта памяти

Функция цели определяется как минимизация резервных компонентов матрицы памяти (S – spare), необходимых для восстановления ее работоспособности в процессе функционирования цифровой системы на кристалле путем синтеза ДНФ покрытия дефектных элементов с последующим выбором минимального конъюнктивного терма , удовлетворяющего ограничениям по числу резервных строк и столбцов , входящих в состав логического произведения:



где каждый результирующий конъюнктивный терм функции Y составлен из идентификаторов строк и столбцов , покрывающих все дефекты в матрице памяти. Лучшее решение есть терм минимальной длины по Квайну, в котором содержатся как строки, так и столбцы, покрывающие все дефекты. В частности, решение может не содержать строк (столбцов), когда для ремонта памяти достаточно только существующих столбцов (строк) из резерва матрицы памяти. Модель процесса определения минимального числа резервных компонентов, покрывающих все обнаруженные дефекты в матрице памяти, сводится к следующим пунктам:



1. Преобразование двумерной модели дефектов матрицы памяти в таблицу покрытия дефектов резервными строками и столбцами матрицы.

Для достижения поставленной цели рассматривается топологическая модель памяти в виде матрицы, идентифицирующей обнаруженные дефекты:

(2.34)



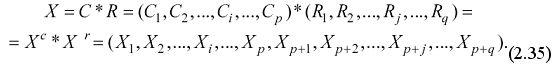
Здесь координата матрицы отмечается 1, если функция исправного поведения ячейки на тесте дает единичное значение, то координата идентифицируется дефектной. После фиксации всех дефектов выполняется построение таблицы покрытия дефектов, где столбцы соответствуют множеству установленных дефектов m, а строки есть номера столбцов и строк матрицы памяти, которые имеют дефекты:



(2.35)



Вместо компонентов двумерной метрики C и R используется одномерный вектор, сконкатенированный из двух последовательностей C и R, мощность которого равна n = p + q:



При этом между элементами исходных наборов (C, R) и результирующим вектором Х существует взаимно однозначное соответствие, установленное в первом столбце матрицы Y. Следует заметить, что преобразование X = C \* R выполняется лишь для удобства рассмотрения и последующего построения ДНФ в рамках единообразия переменных, формирующих булеву функцию. Если данную процедуру не выполнять, то функция будет определена на двух типах переменных, содержащих столбцы и строки матрицы памяти.

2. Построение КНФ для аналитического, полного и точного решения задачи покрытия.

После формирования матрицы покрытия, содержащей нулевые и единичные координаты, выполняется синтез аналитической формы покрытия путем записи КНФ по столбцам. Здесь число конъюнктивных термов, равно количеству столбцов таблицы, а каждая дизъюнкция записывается по единичным значениям рассматриваемого столбца:



Из последнего выражения видно, что каждый столбец имеет только две координаты, имеющие единичное значение, а число логических произведений равно общему числу дефектов m, обнаруженных в матрице памяти.

3. Преобразование КНФ к ДНФ.

Данное преобразование даёт возможность увидеть все решения задачи покрытия. Для этого к КНФ необходимо применить операцию логического умножения и правила минимизации (поглощения) для получения ДНФ:



Здесь представлена обобщенная запись ДНФ, где в пределе число термов равно , n – число строк в обобщенном множестве (C, R) или количество переменных Х в матрице Y, на множестве которых формируются все решения – покрытия дефектов резервными компонентами; если при принимает значение нуля, то переменная превращается в несущественную.



4. Выбор минимальных и точных решений задачи покрытия.

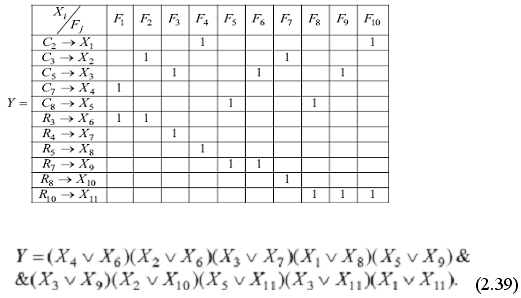
Данный выбор связан с определением конъюнкций минимальной длины в полученной ДНФ. Последующее преобразование к строкам и столбцам матрицы памяти, на основе использования ранее введенного соответствия, даёт возможность записать минимальное покрытие или их совокупность в двумерной метрике строк и столбцов, отвечающие условиям (ограничениям) функции цели на количество резервных компонентов.

Далее предлагается иллюстрация модели процесса восстановления работоспособности матрицы памяти в части определения минимального числа резервных компонентов, покрывающих все дефекты. Матрица памяти с дефектами и резервом [11] представлена на рис. 2.3.

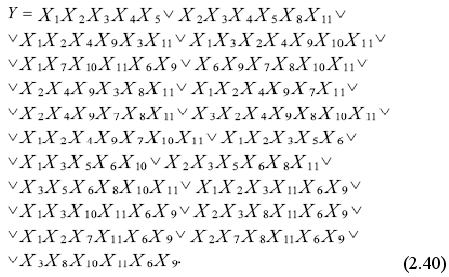
Матрица имеет ограничения на возможность диагностирования и восстановление работоспособности десяти дефектных ячеек, которые определяются двумя строками и пятью столбцами. В соответствии с пунктом 1 модели процесса определения минимального числа резервных компонентов, покрывающих все обнаруженные дефекты в матрице памяти, строится таблица покрытия десяти дефектов одиннадцатью строками, представленными в виде конкатенации подмножеств С и R, находящихся во взаимно-однозначном соответствии с вектором переменных Х:



Далее, в соответствии с таблицей покрытия выполняется построение КНФ, термы которой записаны по единичным значениям столбцов.



Последующие преобразования, связанные с получением ДНФ основываются на применении законов и тождеств булевой алгебры, которые позволяют выполнить логическое перемножение всех десяти сомножителей, последующую минимизацию термов ДНФ путем применения оператора сограней, аксиом поглощения, исключения одинаковых термов. Опустив промежуточные вычисления, окончательный результат представим в следующем виде:



Выбор термов минимальной длины, содержащих 5 переменных, формирует множество оптимальных (минимальных) решений, имеющих вид:

(2.41)



Трансформирование полученной функции к покрытию, содержащему обозначения переменных в виде строк и столбцов матрицы памяти, позволяет представить решения в следующей форме:

(2.41)



Все полученные минимальные решения удовлетворяют требованиям по ограничениям на число резервных компонентов, определенных числами:

(2.42)



Другие решения, определенные в ДНФ, не представляют интереса, поскольку они имеют неоптимальное покрытие дефектных ячеек, определяемое числом переменных (строки + столбцы) в термах более пяти. Последующая технология встроенного ремонта дефектных ячеек заключается в электрическом перепрограммировании дешифратора адреса столбца или строки матрицы памяти. Применительно к памяти, изображенной на рис. 2.3, процедура записи или считывания информации при обращении к любой ячейке столбца 2 будет переадресована к резервному столбцу 11. Соответственно последнему полученному решению в виде первого терма ДНФ функции Y, будут заменены и другие дефектные столбцы на исправные из резерва памяти: 3 – на 12; 5 – на 13; 7 – на 14, 8 – на 15.

Вычислительная сложность АЛМ восстановления работоспособности в части решения задачи покрытия определяется следующим выражением:

(2.43)



где – затраты, связанные с синтезом ДНФ путем логического перемножения исключительно двухкомпонентных дизъюнкций (координата дефекта определяется номером строки и столбца), число которых равно количеству дефектных ячеек; – верхняя граница вычислительных затрат, необходимых для минимизации полученной ДНФ на предельном множестве переменных, равном суммарному числу строк и столбцов



В худшем случае, когда координаты всех дефектных ячеек по строкам и столбцам не коррелированы, – уникальны, например, диагональные дефекты, вычислительная сложность матричного метода становится зависимой только от числа дефектных ячеек, а ее аналитическая запись трансформируется к следующему виду:

(2.44)



Если вместо мощности множества дефектов записать их число, равное m, тогда предыдущее выражение представляется в более простой форме:

(2.45)



Согласно технологии встроенного сервисного обслуживания функциональных модулей цифровых систем на кристаллах, матричный метод восстановления работоспособности на основе решения задачи покрытия имплементируется в кристалл в качестве одного из компонентов I-IP (Infrastructure Intellectual Property), нацеленного на поддержание работоспособности матричной памяти SoC.

Выводы

SoC-микросхемы, в ближайшем будущем, будут составлять более 90% объема кристалла, ориентированного на использование гибких программных средств в ГАС.

Актуальной представляется разработка не только средств быстрого и точного диагностирования, но и создание технологий для осуществления ремонта дефектных ячеек, встроенными средствами сервисного обслуживания в реальном времени и на всех стадиях жизненного цикла изделия. Это позволит существенно уменьшить число выводов чипа, повысить выход годной продукции, уменьшить время выхода изделия на рынок, сократить затраты на сервисное обслуживание, а также исключить внешние средства диагностирования и ремонта.

В процессе написании квалификационной работы бакалавра, была достигнута поставленная цель работы, посредством выполнения следующих этапов:

рассмотрен АЛМ и, соответствующий ему, алгоритм встроенного диагностирования дефектов в функциональных блоках SoC, использующий предварительный анализ ТН в целях уменьшения её объема и последующих вычислений, связанных с построением ДНФ, формирующей все решения по установлению диагноза функциональностей SoC;

выполнена сокращенная инфраструктура сервисного обслуживания функциональностей SoC, отличающаяся минимальным набором процессов встроенного диагностирования в реальном масштабе времени и дает возможность осуществлять сервисы:

тестирование штатных функций на основе генерируемых входных последовательностей ATPG и анализ выходных реакций;

диагностирование с заданной глубиной поиска дефектов путем использования мультизонда стандарта IEEE 1500;

моделирование (Fault Simulation) неисправностей в целях обеспечения выполнения первых двух процедур на основе ТН FDT;

описана мэппинг-модель процесса синтеза дедуктивной структуры, отличающаяся использованием библиотеки дедуктивных компонентов, покрывающих все стандартизованные конструктивы функциональностей, которыми оперирует разработчик, что дает возможность создавать в автоматизированном режиме дедуктивную модель функциональностей цифровой системы на кристалле;

описана мэппинг-модель процесса синтеза тестов, отличающаяся использованием библиотеки встроенных тестовых генераторов для функциональностей DSP SoC, что дает возможность существенно уменьшить время построения тестов, предназначенных для верификации функциональностей и проверки дефектов.

АЛМ восстановления работоспособности памяти основывается на решении задачи покрытия дефектных ячеек резервными элементами путем использования аппарата булевой алгебры. Метод имеет квадратичную вычислительную сложность и может быть аппаратурно реализован как в программном исполнении за пределами кристалла, так и внутри него в виде дополнительного сервисного модуля коррекции дефектов, позволяющего автоматически выполнять восстановление работоспособности элементов памяти в процессе функционирования.

Также в работе рассмотрено априорное задание таблицы неисправностей в виде булевой функции, с точки зрения компактности, которая на конкретном ВЭП трансформируется в компактную запись, определяющую термы ДНФ, как все возможные решения неисправных компонентов, подлежащих ремонту.

Перечень ссылок

1. Хаханов В.И., Хаханова А.В., Литвинова Е.И. Алгебро-логический метод ремонта встроенной памяти SoC // Відмовостійкі системи. – №1. – 2008. – С. 99 – 109.

2. Хаханов В.И., Хаханова И.В. VHDL + Verilog = Синтез за минуты. Харьков: СМИТ, 2007. – 264 с.

3. Zorian Y. What is Infrustructure IP // IEEE Design & Test of Computers. – May – June 2002. P. 5 – 7.

4. Zorian Y., Gizopoulos D. Gest editors’ introduction: Design for Yield and reliability // IEEE Design & Test of Computers. – May – June 2004. – P. 177 – 182.

5. Rashinkar P., Paterson P., Singh L. System-on-chip Verification: Methodology and Techniques, – Kluwer Academic Publishers, 2002. – 393 р.

6. Хаханов В.И. Инфраструктура диагностического обслуживания SoC // Вестник Томского государственного университета. – №4(5). – 2008. – С. 74 – 101.

7. IEEE 1500 Web Site. [Электрон. ресурс]. – Режим доступа: – http: // grouper.ieee.org/groups/1500/.

8. Автоматизация диагностирования электронных устройств / Ю.В. Малышенко и др. / Под ред. В.П. Чипулиса. – М.: Энергоатомиздат, 1986. – 304 с.

9. Shoukourian S., Vardanian V., Zorian Y. SoC Yield Optimization via an Embedded-Memory Test and Repair Infrastructure // IEEE Design and Test of Computers. – 2004. – P. 200 – 207.

10. Zorian Y., Shoukourian S. Embedded-Memory Test and Repair: Infrastructure IP for SoC Yield // IEEE Design and Test of Computers. – 2003. – P. 58 – 66.

11. Парфентий А.Н., Хаханов В.И., Литвинова Е.И. Модели инфраструктуры сервисного обслуживания цифровых систем на кристаллах // АСУ и приборы автоматики. – Вып. 138. – 2007. – С. 83 – 99.

12. Hahanov V., Kteaman H., Ghribi W., Fomina E. HEDEFS – Hardware embedded deductive fault simulation // Proc. volume from the 3-rd IFAC Workshop, Rydzyna, Poland. – 2006. – P. 25 – 29.

13. Youngs L., Paramanandam S. Mapping and Repairing Embedded-Memory Defects // IEEE Design and Test of Computers. – 1997. – P. 18 – 24.

14. Bergeron J. Writing testbenches: functional verification of HDL models. –Springer, 2003. – 512 р.

15. DaSilva F., Zorian Y., Whetsel L. Overview of the IEEE P1500 Standard // ITC International Test Conference. – 2003. – P. 988 – 997.

16. Rossen K. Discrete Mathematics and its Applications. – McGraw Hill, 2003. – 824 p.

17. Бондаренко М.Ф., Кривуля Г.Ф., Рябцев В.Г., Фрадков С.А., Хаханов В.И. Проектирование и диагностика компьютерных систем и сетей. – К.: НМЦ ВО, 2000. – 306 c.

18. Zhong Y., Dropsho S.G., Shen X., Studer A., Ding C. Miss Rate Prediction Across Program Inputs and Cache Configurations // IEEE Trans. on Computers. – 2007. – P. 328 – 343.

19. Hamdioui S., Gaydadjiev G.N., Van de Goor A.J. The State-of-the-art and Future Trends in Testing Embedded Memories // Records IEEE Intern. Workshop on Memory Technology, Design, and Testing, San Jose, CA. – August 2004. – P. 54 – 59.

20. IEEE-1800. IEEE Standard for System Verilog Language. – 2005. – 586 p.

21. Densmore D., Passerone R., Sangiovanni-Vincentelli A. A Platform-Based taxonomy for ESL design // Design & Test of computers. – September – October 2006. – P. 359 – 373.

22. Методические указания к дипломному проекту для студентов специальности 8.091402 «Гибкие компъютерные системы и робототехника» Упоряд. В.В. Токарев, О.М. Цимбал. – Харьков: ХНУРЭ, 2003. – 40 с.

23. Державний стандарт України. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. Чинний від 01.01.96. – К.:Держстантдарт, 1995. – 60 с.

24. ГОСТ 2.105 – 2001. Единая система конструкторской документации. Общие требования к текстовым документам. – М.: Из-во стантдартов, 2001. – 76 с.

25. Единая система конструкторской документации: Справ. пособ. / С.С. Борушек А.А. Волков, М.М. Ефимова и др. - 2-е изд., перераб. и доп. – М.: Изд-во стандартов, 1989. – 352 с.