УДК 658.512.011:681.326:519.713

ИВ ТИЕКУРА (YVES TIECOURA), В.И. ХАХАНОВ, С.В. ЧУМАЧЕНКО, Е.И. ЛИТВИНОВА

АНАЛИЗ КИБЕРПРОСТРАНСТВА И ДИАГНОСТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ МОДУЛЕЙ

Описываются инфраструктура и технологии анализа киберпространства, в рамках которого созданы транзакционная граф-модель и метод диагностирования цифровых систем на кристаллах, ориентированные на существенное уменьшение времени поиска дефектов и затрат памяти для хранения матрицы диагностирования путем формирования тернарных отношений в форме тест, монитор, функциональный компонент. Решаются задачи: создание модели цифровой системы в виде транзакционного графа, а также тернарные матрицы активизации функциональных компонентов на тестах относительно выбранного множества мониторов; разработки метода анализа матрицы активизации для поиска дефектов с заданной глубиной и синтеза логических функций для последующей реализации встроенного аппаратного диагностирования дефектов.

1. Анализ киберпространства и актуальные задачи

Чтобы идея материализовалась и завоевала весь мир, она должна быть простой и понятной каждому. Такого рода идея подняла на вершину успеха компании Microsoft, Google, Kaspersky lab, Intel, IBM. Рождается новая персональная или индивидуальная модель киберпространства – виртуальный персональный киберкомпьютер – Virtual Personal Cybercomputer (VPC), родителями которого можно считать с одной стороны реальные нанотехнологии и цифровые системы на кристаллах, а с другой – виртуальные сервисы по хранению, обработке и приему-передаче информации. Мотивация появления данной модели на рынке определяется: 1) Необходимостью создания «индивидуального или личного виртуального компьютера или пространства», которое нельзя потерять или украсть. 2) Нежеланием пользователя и высокой стоимостью дублирования информации и сервисов при наличии у него нескольких гаджетов (планшет, смартфон, ноутбук). 3) Высоким уровнем надежности и информационной безопасности хранения личных данных и сервисов в течение всей жизни пользователя. Перечисленные условия может обеспечить только индивидуальная ячейка в «швейцарском банке», которая должна и будет создаваться в ближайшие три года для каждого человека планеты как Personal Cyberspace Cell (PCC). Две точки зрения в своем развитии сходятся к одному понятию, PCC = VPC, первая исходит со стороны киберпространства (компьютерных наук), вторая – со стороны персонального компьютера (компьютерной инженерии). Практически при полном отсутствии недостатков PCC имеет следующие преимущества: 1) Инвариантность функционирования по отношению к любому аппаратному интерфейсу, соединяющему пользователя с киберпространством. 2) Дружественность и интеллектуальная адаптивность к «хозяину» по формату 24/7 на протяжении всей жизни. 3) Аутентификация пользователя и PCC по отношению к облачным и другим сервисам киберпространства, которая сегодня завязана на конкретную аппаратуру. 4) Надежность и доступность, сохраняемость и безопасность PCC, переносимость и физическая неуязвимость благодаря своей виртуальности. 5) Эффективная реляционная структуризация данных и сервисов с признаками интеллекта для поиска, распознавания и принятия решений. 6) Высокая рыночная привлекательность создания кибербанков и PCC-форматов (шаблонов, стандартов), которые ориентированы и необходимы каждому человеку планеты (в денежном эквиваленте это составляет сотни миллиардов долларов). 7) Возможность создания прототипа виртуального персонального киберкомпьютера ограниченными силами нескольких раскрученных компаний, имеющих выход на World Market, и двумя-тремя университетами. 8) Ориентировочная стоимость таких работ с созданием начальной инфраструктуры банков киберпространства – 0,5 – 1,5 млрд долларов.

Цель – создание индивидуального и виртуального компьютера в киберпространстве для выполнения интеллектуальных транзакций с данными и сервисами, ориентированными на каждого человека.

Задачи: 1) Определение технической инфраструктуры для функционирования виртуального РСС. 2) Создание структурированной базы данных для хранения информации и сервисов. 3) Разработка шаблона РСС в виде набора взаимосвязанных сервисов и инструментов, ориентированных на удовлетворение потребностей пользователя. 4) Разработка системы защиты персонального киберпространства, данных и сервисов, включающей аутентификацию, ключи, цифровую подпись, криптографию. 5) Создание интеллектуальных средств поиска, распознавания и принятия решений в виде совокупности фильтров, ориентированных на конкретного пользователя. 6) Создание прототипа РСС и его тестирование для различных типов пользователей. 7) Предложение прототипа компаниям, которые имеют выход на рынок электронных технологий, а также Public Relations путем выступлений в Internet, TV, на конференциях и семинарах.

Сущность – создание инфраструктуры оптимальной организации индивидуального киберпространства в виде виртуального компьютера, имеющего следующие сервисы: 1) электронная почта и телефония; 2) Internet-браузеры для поиска, распознавания и принятия решений; 3) аудио- и видеопроигрыватели; 4) текстовые и звуковые редакторы; 5) электронный banking и shopping; 6) индивидуальный бизнес-браузер для организации рабочего дня; 7) браузер для организации отдыха, культуры и спорта; 8) traveling-браузер; 9) структурированная реляционная база данных для хранения истории и всех типов данных; 10) внешний интерфейс Public Relations; 11) медицинское обслуживание и сервисы; 12) комплексная система защиты информации и сервисов.

Киберкомпьютер (персональный и виртуальный) – виртуальное отображение в киберпространстве функций персонального компьютера для выполнения интеллектуальных транзакций с данными и сервисами, индивидуально ориентированными на каждого человека. Киберпространство – совокупность взаимодействующих по метрике информационных процессов и явлений, использующих в качестве носителя компьютерные системы и сети. Метрика – способ измерения расстояния в пространстве между компонентами процессов или явлений. Расстояние в киберпространстве – это xor-взаимодействие компонентов процессов или явлений, представленных векторами, которое имеет скалярную проекцию в виде расстояния по Хэммингу. Инвариантами расстояния являются: производная (булева), степень изменения, различия или близости компонентов процесса или явления. Процедуры сравнения, измерения, оценивания, распознавания, тестирования, диагностирования, идентифицирования и принятия решений сводятся к определению xor-отношения, которое для циклических структур равно нуль-вектору.

Далее предлагается решение практической задачи – диагностирование функциональных блоков цифровой системы на кристалле на основе использования аппарата анализа киберпространства.

2. TAB-модель диагностирования дефектных компонентов в SoC

Мотивация определяется рыночной привлекательностью матричного метода поиска дефектов в компонентах (программных и аппаратных) цифровых систем на кристаллах, как самого технологичного, который ориентирован на параллельную обработку данных, что дает возможность существенно уменьшить время диагностического обслуживания при возникновении неисправностей.

Цель исследования – создание модели, метода и их аппаратной реализации, ориентированных на существенное уменьшения времени тестирования и затрат памяти для хранения матрицы диагностирования путем формирования тернарных отношений (тест – монитор – функциональный компонент) внутри одной таблицы или TAB: Tests – Assertions – Blocks.

Задачи: 1) Разработка модели цифровой системы в виде транзакционного графа, а также матрицы активизации функциональных компонентов на тестах относительно выбранного множества мониторов [1-6]. 2) Разработка метода анализа матрицы активизации для поиска дефектов с заданной глубиной [4-7]. 3) Синтез логических функций для встроенного диагностирования дефектов [8-11].

Модель тестирования цифровой системы представлена в виде следующего преобразования начального уравнения диагноза, определенного xor-отношением параметров <тест – функциональность – неисправные блоки >:

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

которое оформлено в тернарное матричное отношение компонентов:

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

Здесь координата матрицы (таблицы) равна 1, если пара тест-монитор $(T \times A)_i$ проверяет (активизирует) дефекты функционального блока $B_j \in B$.

Модель цифровой системы представлена в виде транзакционного графа

$$G =< B, A >, B = \{B_1, B_2, ..., B_i, ..., B_n\}, A = \{A_1, A_2, ..., A_j, ..., A_m\},$$

где определены множество дуг – функциональных блоков и вершин – мониторов для наблюдения совокупности переменных цифровой системы. Для целей диагностирования на графовую модель накладывается совокупность тестовых сегментов $T = \{T_1, T_2, ..., T_r, ..., T_k\}$, которая активизирует транзакционные пути в графе. В общем случае модель тестирования представлена декартовым произведением $M =< B \times A \times T >$, она имеет размерность $Q = n \times m \times k$. Чтобы уменьшить объем диагностической информации, предлагается каждому тесту поставить в соответствие монитор, который отвечает за визуализацию пути активизации функциональных блоков, что дает возможность уменьшить размерность модели (матрицы) до $Q = n \times k$ при сохранении всех возможностей отношения триады $M =< B \times A \times T >$. Для пары тест-монитор возможны не только взаимно-однозначные соответствия $< T_i \to A_j >$, но и функциональные $< \{T_i, T_r\} \to A_j >$, а также инъективные $< T_i \to \{A_j, A_s\} >$. Такое многообразие соответствий дает возможность дублировать один тестовый сегмент для разных мониторов, равно как и нагружать несколько тестов на один и тот же монитор. При этом ячейка матрицы $M_{ij} = \{0,1\}$ всегда сохраняет свою размерность, равную одному биту.

Аналитическая обобщенная модель матричного диагностирования с использованием механизма мониторов ориентирована на достижение заданной глубины поиска дефектов и представлена в следующем виде:

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

Здесь B_i – группа операторов кода, нагруженная на вершину L_i (переменная, регистр, счетчик, память) и формирующая ее состояние; G – функциональность, представленная транзакционным графом $G = (L, A) \times B$ в виде декартова произведения множества вершин и дуг; A – совокупность мониторов, как подмножество вершин транзакционного графа $A \subseteq L$. Метод поиска неисправностей функциональных блоков использует предварительно построенную таблицу (матрицу) активизации ТАФБ $M = [M_{ij}]$, где строка есть отношение между тестовым сегментом и подмножеством активизированных блоков

$$T_i \to A_j \approx (M_{i1}, M_{i2}, ..., M_{ij}, ..., M_{in}), M_{ij} = \{0,1\},$$

наблюдаемых на мониторе A_j. Столбец таблицы формирует отношение между функциональным блоком, тестовыми сегментами и мониторами $M_j = B_j \approx f(T, A)$. В механизм мониторов может быть введен параметр модельного времени, который частично усложняет матрицу активизации, указывая временной или модельный такт, на котором выполняет-

ся мониторинг состояния вершины или функционального блока на тест-сегменте $A_j = f(T_i, B_j, t_j)$.

Для диагностирования неисправностей на стадии моделирования определяется обобщенная реакция (вектор-столбец) $m = \{m_1, m_2, ..., m_i, ..., m_p\}$ механизма мониторов A на тест-сегменты T путем формирования $m_i = f(T_i, A_i)$. Поиск неисправного ФБ основан на определении xor-операции между вектором состояния ассерций и столбцов таблицы ФН $m \oplus (M_1 \vee M_2 \vee ... \vee M_j \vee ... \vee M_n)$. Выбор решения определяется методом xor-анализа столбцов путем выбора совокупности векторов B_j с минимальным числом единичных координат

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

формирующих функциональные блоки с неисправностями, проверяемыми на тестовых сегментах. В дополнение к модели матричного диагностирования необходимо описать следующие важные свойства матрицы:

$$1)\ M_i = (T_i - A_j);$$

$$2)\ \vee_{i=1}^{m} M_{ij} \to \forall M_j = 1;$$

$$3)\ M_{ij} \oplus_{j=1}^{n} M_{rj} \neq M_{ij};$$

$$4)\ M_{ij} \oplus_{i=1}^{k} M_{ir} \neq M_{ij};$$

$$5)\ \log_2 n \le k \leftrightarrow \log_2 |B| \le |T|$$

$$6)\ B_j = f(T, A) \to B \oplus T \oplus A = 0.$$

Свойства означают: 1) Каждая строка матрицы есть соответствие или подмножество декартова произведения (тест-монитор). 2) Дизъюнкция всех строк матрицы дает вектор, равный единицам по всем координатам. 3) Все строки матрицы различны, что исключает тестовую избыточность. 4) Все столбцы матрицы различны, что исключает существование эквивалентных неисправностей. 5) Число строк матрицы должно быть больше двоичного логарифма от числа столбцов, что определяет потенциальную диагностируемость всех блоков. 6) Функция диагностирования блока зависит от совокупного теста и мониторов, которые должны быть минимизированы без нарушения диагнозопригодности.

3. Диагнозопригодность проекта

Что касается качества модели диагностирования функциональных нарушений, то она показывает эффективность использования пары (тест, ассерции) для заданной глубины диагностирования. Оценка качества модели функционально зависит от длины теста $|T|$, числа ассерций наблюдения $|A|$, количества распознаваемых блоков с функциональными нарушениями N_d на общем числе программных блоков N:

$$Q = E \times D = \frac{]\log_2 N[}{|T| \times |A|} \times \frac{N_d}{N}.$$

Эффективность диагностирования есть отношение минимального числа двоичных разрядов, необходимых для идентификации (распознавания) всех блоков, к реальному количеству разрядов кода, представленному произведением длины теста на число ассерций в каждом из них. Если первая дробь оценки равна 1 и все блоки с ФН распознаются ($N_d = N$), то тест и ассерции оптимальны, что доставляет критерию качества модели диагностирования значения, равного 1. Далее рассмотрим пример матрицы ABC-графа,

который представлен на рис. 1 и имеет 14 функциональных блоков – дуг, а также 9 мониторов – вершин.

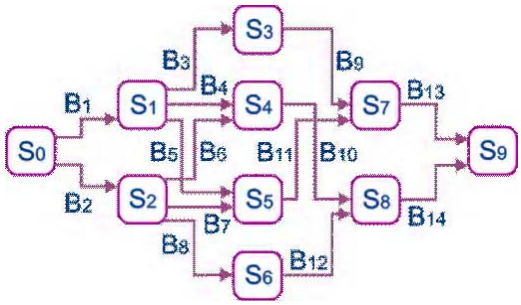

Рис. 1. Пример ABC-графа транзакций

Для такого графа существует 2 решения при создании модели диагностирования дефектов с одной и тремя ассерциями или мониторами:

$$Q_1 = \frac{]\log_2 14[}{|6|\times|1|}\times\frac{10}{14} = 0{,}5;$$

$$Q_2 = \frac{]\log_2 14[}{|6|\times|3|}\times\frac{14}{14} = 0{,}2.$$

Несмотря на то, что качество модели лучше в первом варианте (из-за меньшего объема таблицы активизации), вторая модель – более предпочтительна, поскольку она имеет максимальную глубину диагностирования, когда все 14 блоков распознаются за счет добавления двух ассерций. Оценка позволяет определить минимальные затраты по длине теста и числу ассерций для создания модели с максимальной глубиной диагностирования.

Интерес представляет оценивание качества структуры кода проекта с позиции диагностируемости (diagnosability) блоков программного продукта. Цель анализа – определить количественную оценку структуры графа и места (вершины) для установки ассерционных мониторов, создающих максимальную глубину диагностирования функциональных нарушений программных блоков. Здесь важна не управляемость и наблюдаемость, как в тестопригодности (testability), а различимость программных блоков с функциональными нарушениями, в пределе представляющая ноль блоков с эквивалентными (неразличимыми) нарушениями. Такая оценка может быть полезной для сравнения графов, реализующих одинаковую функциональность. Здесь необходимо оценивать структуру графа с позиции потенциально заложенной в нем глубины поиска функциональных нарушений программного продукта. Возможным вариантом может быть диагнозопригодность ABC-графа как функция, зависящая от таких смежных дуг при каждой вершине (формирующих число N_n), одна из которых – входящая, другая – исходящая. Такие дуги составляют пути без схождений и разветвлений (N – общее количество дуг в графе):

$$D = \frac{N - N_n}{N}.$$

Каждая вершина, объединяющая 2 дуги, которые входят в число N_n, называется транзитной. Оценка N_n есть число неразличимых функциональных нарушений программных блоков. Места потенциальной установки мониторов для различения ФН – транзитные вершины. С учетом приведенной оценки диагнозопригодности D качество модели диагностирования программного продукта принимает вид:

$$Q = E \times D = \frac{]\log_2 N[}{|T|\times|A|}\times\frac{N - N_n}{N}.$$

Правила синтеза диагнозопригодных программных продуктов: 1) Тест (testbench) должен создавать минимальное число одномерных путей активизации, покрывающих все вершины и дуги ABC-графа. 2) Базовое число мониторов-ассерций равно количеству конечных вершин графа, не имеющих исходящих дуг. 3) В каждой вершине, имеющей одну

входную и одну выходную дугу, может быть размещен дополнительный монитор. 4) Параллельно-независимые блоки кода имеют n мониторов и один тест или один объединенный монитор и n тестов. 5) Последовательно соединенные блоки имеют 1 тест активизации последовательного пути и n-1 монитор или n тестов и n мониторов. 6) Вершины графа, имеющие различное число входных и выходных дуг, создают условия для диагностируемости данного участка благодаря одномерным тестам активизации без установки дополнительных мониторов. 7) Совокупность тестовых сегментов (testbench) должна составлять 100%-ное покрытие функциональных режимов (functional coverage), заданных вершинами ABC-графа. 8) Функция диагнозопригодности прямо пропорциональна длине теста, числу ассерций и обратно пропорциональна двоичному логарифму от числа программных блоков:

$$D = \frac{N - N_n}{N} = f(T, A, N) = \frac{|T| \times |A|}{]\log_2 N[}.$$

Диагнозопригодность как функция, зависящая от структуры графа (программного продукта), теста и ассерционных мониторов всегда может быть приведена к единичному значению. Для этого существует два альтернативных пути. Первый – увеличение тестовых сегментов, активизирующих новые пути, для различения эквивалентных неисправностей без наращивания ассерций, если структура графа программных блоков имеет такой потенциал связей. Второй – размещение дополнительных ассерционных мониторов в транзитных вершинах графа. Возможен и третий, гибридный вариант, основанный на совместном применении двух перечисленных выше путей. Отношение трех компонентов (число программных блоков, мощность механизма ассерций и длина теста) при единичном значении качества модели диагностирования и диагнозопригодности формирует плоскость оптимальных решений $D = 1 \to \frac{|T| \times |A|}{]\log_2 N[} = 1 \to]\log_2 N[= |T| \times |A|$. Она может быть полезной для выбора квазиоптимального варианта альтернативного пути достижения полной различимости на паре $|T| \times |A|$ функциональных нарушений программных блоков.

4. Мультиуровневая модель и метод (движок) диагностирования цифровой системы

Представлена мультидеревом B, где каждая вершина есть трехмерная таблица активизации функциональных модулей, а дуги, исходящие из нее, есть переходы на нижний уровень детализации при диагностировании, когда замена рассматриваемого неисправного функционального блока слишком дорога:

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

n – число уровней мультидерева диагностирования; m_r – количество функциональных блоков или компонентов на уровне r; k_{rs} – число компонентов в таблице B^{rs}; $B_{ij}^{rs} = \{0,1\}$ – компонент таблицы активизации, определенный сигналами проверки (непроверки) функционального модуля тест-сегментом $T_{i\text{-}A_j}$ относительно наблюдаемого монитора A_i. Каждая вершина-таблица имеет число исходящих вниз дуг, равное количеству функциональных блоков, диагностируемых (представленных) в таблице активизации. Структура мультидерева, соответствующая модели многоуровневого диагностирования, представлена на рис. 2.

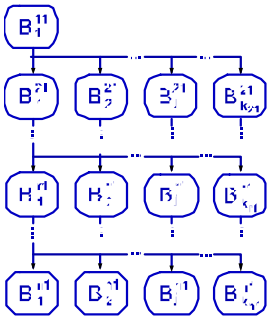

Рис. 2. Фрагмент мультидерева диагностирования цифровых систем

Процесс-модель или метод поиска дефектов по мультидереву диагностирования сводится к созданию движка (рис. 3) для спуска по одной из ветвей дерева на такую глубину, которая удовлетворяет пользователя по степени детализации:

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

Здесь выполняется векторная xor-операция между столбцами матрицы и вектором экспериментальной проверки A^{rs}, который определяется реакцией функциональности, снятой с мониторов (ассерции или разряды регистра граничного сканирования) при подаче всех тест-сегментов. Если хотя бы одна координата полученной векторной xor-суммы равна нулю $B_j^{rs} \oplus A^{rs} = 0$, то выполняется одно из действий: переход к матрице активизации нижнего уровня $B_j^{r+1,s}$ или восстановление работоспособности функционального блока B_j^{rs}. При этом анализируется, что важнее: 1) время – тогда выполняется ремонт рассматриваемого блока с неисправностью; 2) деньги – тогда осуществляется переход вниз, для уточнения места дефекта, поскольку замена более мелкого блока существенно уменьшает стоимость ремонта. Если хотя бы одна координата полученного вектора xor-суммы равна единице $B_j^{rs} \oplus A^{rs} = 1$, то выполняется переход к анализу следующего столбца матрицы. При нулевых значениях всех координат вектора (ассерционных) мониторов $A^{rs} = 0$ фиксируется исправное состояние всего изделия. Если в рассматриваемой таблице зафиксированы все векторные xor-суммы, не равные нулю $B_j^{rs} \oplus A^{rs} = 1$, то коррекции подлежит тест, построенный для проверки данной функциональности.

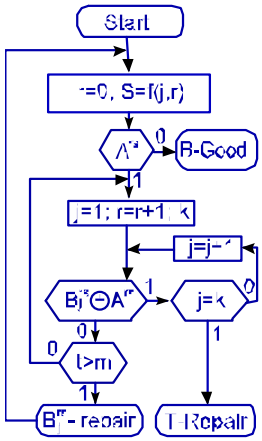

Рис. 3. Движок обхода мультидерева диагностирования

Таким образом, представленная на рис. граф-схема дает возможность эффективно осуществлять сервисное обслуживание сколь угодно сложной технической системы. Преимущества такого движка, инвариантного к уровням иерархии, заключаются в простоте подготовки и представления диагностической информации в виде минимизированной таблицы активизации функциональных блоков на тестовых сегментах.

5. Верификация моделей и метода диагностирования

Для пояснения работоспособности модели и метода далее рассмотрим функциональности трех модулей, входящих в состав цифрового фильтра Добеши [11]. Первым является компонент Row_buffer, для которого создан транзакционный граф на основе RTL-модели (рис. 4). Вершины представлены состояниями переменных и мониторов, отвечающих за входящие в вершину транзакции или дуги, которым соответствуют функциональные блоки.

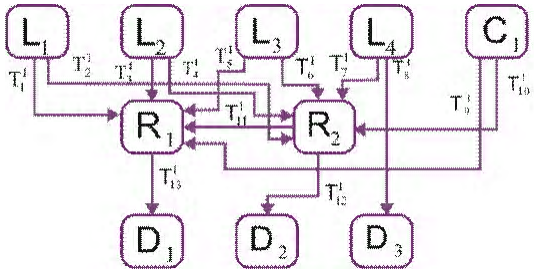

Рис. 4. Компонент Row_buffer транзакционного графа

На основе графа, полученного в процессе моделирования, строится таблица активизации функциональных блоков, строки которой представляют пути активизации блоков к заказанной вершине-монитору. Таблица представляет собой покрытие строками-путями всех столбцов или функциональных блоков. При этом в ней не должно быть хотя бы двух одинаковых столбцов. Отличие таблицы заключается в формировании пары <тест – наблюдаемая вершина>, что дает возможность существенно сократить размерность таблицы при 100% распознавании всех дефектных блоков. Здесь самое главное отличие предложенной модели заключается в возможности описания с помощью таблицы следующих отношений: различные тесты – одна вершина; один тест – различные вершины:

A_{ij}	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}
$t_1 \to D_3$	.	.	.	.	.	.	.	1	.	.	.	.	.
$t_2 \to D_1$	1	.	.	.	.	.	.	.	.	.	.	.	1
$t_3 \to D_1$	.	.	1	.	.	.	.	.	.	.	.	.	1
$t_4 \to D_1$	.	.	.	.	1	.	.	.	.	.	.	.	1
$t_5 \to D_1$	.	.	.	.	.	.	.	.	.	.	1	.	1
$t_6 \to D_1$	.	.	.	.	.	.	.	.	1	.	.	.	1
$t_7 \to D_2$	.	1	.	.	.	.	.	.	.	.	.	1	.
$t_8 \to D_2$	.	.	.	1	.	.	.	.	.	.	.	1	.
$t_9 \to D_2$	.	.	.	.	.	1	.	.	.	.	.	1	.
$t_{10} \to D_2$	.	.	.	.	.	.	1	.	.	.	.	1	.
$t_{11} \to D_2$	.	.	.	.	.	.	.	.	.	1	.	1	.

С помощью матрицы активизации функциональных блоков (транзакционного графа) и xor-метода поиска дефектов достаточно просто синтезировать логические функции для формирования комбинационной схемы, определяющей в процессе и по результатам моделирования номер функционального блока, который имеет семантические ошибки:

$$D_3 = T_8^1;$$

$$D_1 = T_{13}^1 T_1^1 \vee T_{13}^1 T_3^1 \vee T_{13}^1 T_5^1 \vee T_{13}^1 T_{11}^1 \vee T_{13}^1 T_9^1;$$

$$D_2 = T_{12}^1 T_2^1 \vee T_{12}^1 T_4^1 \vee T_{12}^1 T_6^1 \vee T_{12}^1 T_7^1 \vee T_{12}^1 T_{10}^1.$$

Такое свойство становится возможным благодаря отсутствию эквивалентных неисправностей или одинаковых столбцов в матрице активизации. Поэтому фиксация фактического состояния всех мониторов в вершинах D_1, D_2, D_3 на 11 тестовых наборах дает возможность однозначно идентифицировать некорректный функциональный модуль путем выполнения xor-операции между вектором ассерций и столбцами матрицы активизации. Нулевое значение всех координат результата xor-операции определяет номер столбца, соответствующего неисправному модулю. Имплементация модели и метода в логическую функцию дает возможность определять неисправный блок еще до завершения диагностического эксперимента, если это возможно. Это означает существенную экономию времени

диагностирования отдельных видов дефектов. Например, тест-монитор $t_1 \to D_3$ дает возможность идентифицировать уже на первом тесте неисправность блока B_8.

В качестве второго тестового примера для практического использования разработанной модели активизации и xor-метода поиска дефектов далее предлагается синтез матрицы диагностирования для модуля дискретного косинусного преобразования из Xilinx библиотеки, фрагмент которого представлен листингом 1.

Листинг 1. Фрагмент функционального покрытия

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

Для всех 12 модулей фильтра разработаны транзакционные графы, таблицы активизации и логические функции для тестирования и поиска дефектов дискретного косинусного преобразования. Граф с матрицей активизации и логической функцией (рис. 5), также принадлежащей фильтру, представлены ниже.

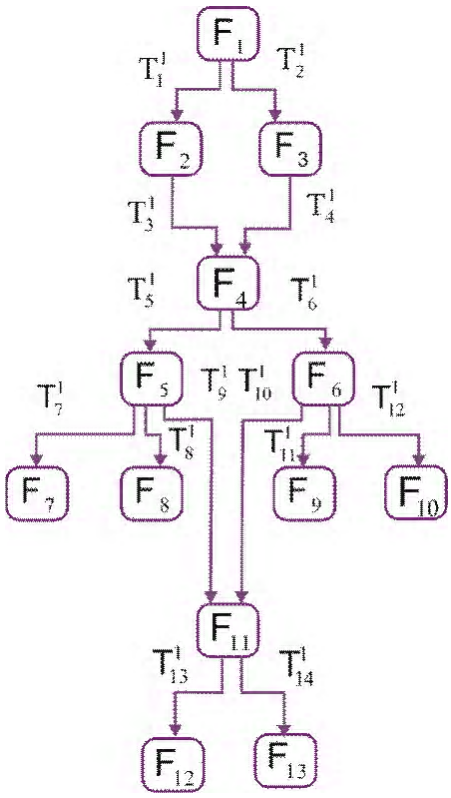

Рис. 5. Транзакционный граф main-RTL

Данному графу ставится в соответствие следующая матрица диагностирования:

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

Система функций диагностирования:

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

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

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

Фрагмент механизма мониторов представлен листингом 2.

Листинг 2. Фрагмент кода механизма мониторов

```
sequence first( reg[7:0] a, reg[7:0]b);
 reg[7:0] d;
 (!RST,d=a)
##7 (b==d);
  endsequence
property f(a,b);
 @(posedge CLK)
 // disable iff(RST||$isunknown(a)) first(a,b);
!RST |=> first(a,b);
endproperty
odin:assert property (f(xin,xa7_in))
 // $display("Very good");
else $error("The end, xin =%b,xa7_in=%b", $past(xin, 7),xa7_in);
```

В результате тестирования дискретного косинусного преобразования в среде Riviera, Aldec были найдены неточности в семи строках HDL-модели:

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

Последующая коррекция кода привела к листингу 3.

Листинг 3. Исправленный фрагмент кода

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

6. Заключение

1. Представлены инфраструктура и технологии анализа киберпространства, в рамках которого созданы транзакционная граф-модель и метод диагностирования цифровых сис-

тем на кристаллах, ориентированные на существенное уменьшение времени поиска дефектов и затрат памяти для хранения матрицы диагностирования путем формирования тернарных отношений в форме тест, монитор, функциональный компонент.

2. Предложена усовершенствованная процесс-модель определения функциональных нарушений в программном или аппаратном изделии, которая отличается использованием xor-операции, что дает возможность повысить быстродействие диагностирования одиночных или кратных дефектов (функциональных нарушений) на основе параллельного анализа таблицы неисправностей, стандарта граничного сканирования IEEE 1500 и векторных операций and, or, xor.

3. Представлена модель диагностирования функциональностей цифровой системы на кристалле в форме мультидерева и метод обхода вершин дерева, имплементированный в движок поиска дефектов с заданной глубиной, которая существенно повышает быстродействие сервисного обслуживания программных и аппаратных компонентов промышленных изделий.

4. Выполнена тестовая верификация метода диагностирования на трех реальных примерах, представленных функциональностями фильтра косинусного преобразования цифровой системы на кристалле, которая показала состоятельность полученных результатов для минимизации времени поиска дефектов и памяти для хранения диагностической информации, а также повышения глубины диагностирования цифровых изделий.

Список литературы: 1. *Основы* технической диагностики / Под. ред. П.П.Пархоменко. М.: Энергия, 1976. 460с. **2.** *Пархоменко П.П., Согомонян Е.С.* Основы технической диагностики (Оптимизация алгоритмов диагностирования, аппаратурные средства) / Под ред. П.П. Пархоменко. М.: Энергия. 1981. 320 с. **3.** *Инфраструктура* мозгоподобных вычислительных процессов / М.Ф. Бондаренко, О.А. Гузь, В.И. Хаханов, Ю.П. Шабанов-Кушнаренко. Харьков: Новое слово. 2010. 160 с. **4.** *Проектирование* и верификация цифровых систем на кристаллах / В.И. Хаханов, И.В. Хаханова, Е.И. Литвинова, О.А. Гузь. Харьков: Новое слово. 2010. 528с. **5.** *Семенец В.В., Хаханова И.В., Хаханов В.И.* Проектирование цифровых систем с использованием языка VHDL. Харьков: ХНУРЭ. 2003. 492 с. **6.** *Хаханов В.И., Хаханова И.В.* VHDL+Verilog = синтез за минуты. Харьков: ХНУРЭ. 2006. 264с. **7.** *IEEE* Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture IEEE Std 1149.7-2009. 985 p. **8.** *Da Silva F., McLaurin T., Waayers T.* The Core Test Wrapper Handbook. Rationale and Application of IEEE Std. 1500™. Springer. 2006. XXIX. 276 p. **9.** *Marinissen E.J., Yervant Zorian.* Guest Editors' Introduction: The Status of IEEE Std 1500 // IEEE Design & Test of Computers. 2009. No26(1). P.6-7. **10.** *Benso A., Di Carlo S., Prinetto P., Zorian Y.* IEEE Standard 1500 Compliance Verification for Embedded Cores // IEEE Trans. VLSI. 2008. No 16(4). P. 397-407. **11.** *Хаханов В.И., Литвинова Е.И., Чумаченко С.В. Гузь О.А.* Логический ассоциативный вычислитель. Электронное моделирование. 2011. № 1. С.73-83.

Поступила в редколлегию 02.06.2011

Тиекура Ив (Tiecoura Yves), аспирант кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

Хаханов Владимир Иванович, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

Чумаченко Светлана Викторовна, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: математическое моделирование, теория рядов, методы дискретной оптимизации. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: ri@kture.kharkov.ua.

Литвинова Евгения Ивановна, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: автоматизация диагностирования и встроенный ремонт компонентов цифровых систем в пакете кристаллов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-421. E-mail: kiu@kture.kharkov.ua.