

МОДЕЛИ ГЕНЕРАЦИИ ТЕСТОВ И МЕТОДЫ ДИАГНОСТИРОВАНИЯ SOC-КОМПОНЕНТОВ

*TIECOURA YVES, NGENE CHRISTOPHER
UMERAN, ЛИТВИНОВА Е.И., ХАХАНОВ В.И.,
АЛЕКСАНДРОВ В.И.*

Предлагается структурная модель отношений компонентов (функциональность, устройство, тест, дефекты), которая позволяет определять и классифицировать пути решения практических задач, включая синтез тестов, моделирование неисправностей и поиск дефектов. Описываются методы синтеза тестов для функциональностей, заданных матричными формами описания поведения цифровых компонентов, которые отличаются параллелизмом векторных операций над таблицами. Предлагаются усовершенствованные методы поиска функциональных нарушений, которые дают возможность существенно повысить быстродействие вычислительных процедур, связанных с диагностированием и восстановлением работоспособности программных и аппаратных продуктов. Модели и методы синтеза тестов для функциональностей и диагностирования нарушений могут быть использованы в качестве встроенных компонентов инфраструктуры сервисного обслуживания цифровых систем на кристаллах с применением стандарта граничного сканирования IEEE 1500.

Введение

Актуальность. Тенденция последних лет в части создания новых коммуникационных, вычислительных и информационных сервисов, полезных для человека, обращает внимание на создание все более специализированных гаджетов (gadget), обладающих существенными преимуществами перед персональными компьютерами и ноутбуками: энергопотребление, компактность, вес, стоимость, функциональные возможности, дружелюбность интерфейса. Практически вся десятка лучших за 2010 год специализированных изделий (Apple iPad, Samsung Galaxy S, Apple MacBook Air, Logitech Revue, Google Nexus One (HTC Desire), Apple iPhone 4, Apple TV, Toshiba Libretto W100, Microsoft Kinect, Nook Color) реализована в виде цифровых систем на кристаллах. Мировой рынок таких гаджетов насчитывает сегодня порядка 3 миллиардов изделий. Для их эффективного проекти-

рования, производства и эксплуатации создаются новые технологии и инфраструктуры сервисного обслуживания. Один из возможных шагов в данном направлении описан ниже.

Цель – разработка новых технологий сервисного обслуживания компонентов цифровых систем на кристаллах, ориентированных на решение задач синтеза тестов и поиска дефектов в программных и аппаратных продуктах.

Задачи: 1) Создание структурной модели отношений компонентов (функциональность, устройство, тест, дефекты) для определения и классификации путей решения практических задач, включая синтез тестов, моделирование неисправностей и поиск дефектов. 2) Разработка методов синтеза тестов для функциональностей, заданных матричными формами описания поведения цифровых компонентов. 3) Усовершенствование метода поиска функциональных нарушений для повышения быстродействия вычислительных процедур диагностирования программных и аппаратных продуктов.

Источники: 1. Модель формулирования задач технической диагностики [1-6]. 2. Синтез тестов для функциональностей [4-8]. 3. Встроенное диагностирование цифровых систем [9-17]. 3. Аппаратура и матричные процессоры [18-21].

1. Модели генерации тестов для функциональных элементов

Предлагаются технологичные и эффективные процесс-модели синтеза тестов и их применение для диагностирования функциональных нарушений в программных и/или аппаратных продуктах. Используются табличные структуры данных, которые ориентированы на параллельное выполнение векторных логических операций при генерации тестов и поиске дефектных компонентов.

Проблема синтеза или анализа произвольной системы может быть сформулирована в виде взаимодействия (симметрической разности) в кибернетическом пространстве ее модели F с входными воздействиями T и реакциями L:

$$f(F, T, L) = \emptyset \rightarrow F \Delta T \Delta L = \emptyset .$$

Кибернетическое пространство (структура) есть совокупность взаимосвязанных компонентов, с задан-

ной адекватностью описывающих процессы или явления для достижения поставленных целей. Каждый компонент представлен n-мерной фигурой, где координата определена в многозначном алфавите, например Кантора $A = \{0, 1, x, \emptyset\}$, $x = \{0, 1\}$. Мерность формирует структуру фигуры: $n=1$ – вектор длины k ; $n=2$ – матрица размерностью $k_1 \times k_2$; $n=3$ – куб размерностью $k_1 \times k_2 \times k_3$.

Метрика β кибернетического или векторного логического двоичного пространства определяется единственным равенством, которое формирует нулевую хог-сумму расстояний между ненулевым и конечным числом точек, замкнутых в цикл:

$$\beta = \bigoplus_{i=1}^n d_i = 0.$$

Иначе: метрика β векторного логического двоичного пространства есть равная нулю хог-сумма расстояний между конечным числом точек, образующих цикл. Сумма n-мерных двоичных векторов, задающих координаты точек циклической фигуры, равна нулю. Данное определение метрики ставит во главу угла не элементы множества, но отношения, что позволяет сократить систему аксиом с трех до одной и распространить ее действие на сколь угодно сложные конструкции n-мерного пространства. Классическое задание метрики для определения взаимодействия одной, двух и трех точек в векторном логическом пространстве является частным случаем β -метрики при $i = 1, 2, 3$ соответственно:

$$M = \begin{cases} d_1 = 0 \leftrightarrow a = b; \\ d_1 \oplus d_2 = 0 \leftrightarrow d(a, b) = d(b, a); \\ d_1 \oplus d_2 \oplus d_3 = 0 \leftrightarrow d(a, b) \oplus d(b, c) = d(a, c). \end{cases}$$

Метрика β векторного логического многозначного пространства есть равная пустому множеству симметрическая разность расстояний между конечным числом точек, образующих цикл:

$$\beta = \Delta_{i=1}^n d_i = \emptyset.$$

Равенство пустому множеству симметрической разности теоретико-множественного взаимодействия подчеркивает равнозначность компонентов, участвующих в формировании уравнения, где единственная операция, используемая в аналитической (четырёхзначной) модели, определяется в виде:

Δ	0	1	x	\emptyset
0	\emptyset	x	1	0
1	x	\emptyset	0	1
x	1	0	\emptyset	x
\emptyset	0	1	x	\emptyset

Для ее определения необходимо выполнить несложные преобразования, представленные в виде следующих доказательств:

$$\begin{aligned} x\Delta 0 &= (x \cap \tilde{0}) \cup (\tilde{x} \cap 0) = (x \cap 1) \cup (\emptyset \cap 0) = 1; \\ x\Delta 1 &= (x \cap \tilde{1}) \cup (\tilde{x} \cap 1) = (x \cap 0) \cup (\emptyset \cap 1) = 0; \\ \emptyset\Delta 0 &= (\emptyset \cap \tilde{0}) \cup (\tilde{\emptyset} \cap 0) = (\emptyset \cap 1) \cup (x \cap 0) = 0; \\ \emptyset\Delta 1 &= (\emptyset \cap \tilde{1}) \cup (\tilde{\emptyset} \cap 1) = (\emptyset \cap 0) \cup (x \cap 1) = 1; \\ 0\Delta 1 &= (0 \cap \tilde{1}) \cup (\tilde{0} \cap 1) = (0 \cap 0) \cup (1 \cap 1) = x; \\ x\Delta \emptyset &= (x \cap \tilde{\emptyset}) \cup (\tilde{x} \cap \emptyset) = (x \cap x) \cup (\emptyset \cap \emptyset) = x; \\ x\Delta x &= (x \cap \tilde{x}) \cup (\tilde{x} \cap \emptyset) = (\emptyset \cap x) \cup (\emptyset \cap \emptyset) = \emptyset. \end{aligned}$$

При переходе к двузначному алфавиту модель синтеза или анализа кибернетической системы трансформируется к виду

$$f(F, T, L) = 0 \rightarrow F \oplus T \oplus L = 0,$$

где фигурирует уже симметричная хог-операция, имеющая таблицу истинности:

\oplus	0	1
0	0	1
1	1	0

Практический интерес представляет обобщение и классификация задач технической диагностики путем создания модели хог-отношений на множестве из четырех основных компонентов. Прimitивные процедуры синтеза тестов, моделирования неисправностей и поиска дефектов можно свести к графу (рис. 1) полного взаимодействия четырех вершин (функциональность, устройство, тест, дефекты) $G = (F, U, T, L)$ на основе хог-операции.

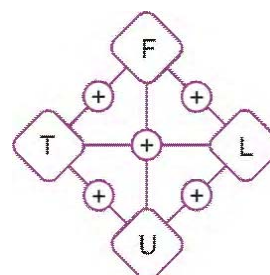


Рис. 1. Граф взаимодействия компонентов технической диагностики

Такой граф порождает четыре базовых треугольника, которые формируют полное множество из 12 практически полезных триад отношений, формулирующих задачи технической диагностики:

$T \oplus F \oplus L = 0$	$T \oplus L \oplus U = 0$	$T \oplus F \oplus U = 0$	$F \oplus L \oplus U = 0$
1) $T = F \oplus L$	4) $T = L \oplus U$	7) $T = F \oplus U$	10) $F = L \oplus U$
2) $F = T \oplus L$	5) $L = T \oplus U$	8) $F = T \oplus U$	11) $L = F \oplus U$
3) $L = T \oplus F$	6) $U = T \oplus L$	9) $U = T \oplus F$	12) $U = F \oplus L$

Содержательно данные задачи можно классифицировать в группы и сформулировать следующим образом.

Группа 1 – теоретические эксперименты (на модели функциональности), без устройства: 1) Синтез теста по модели функциональности для заданного списка неисправностей. 2) Построение модели функциональности на основе заданного теста и списка неисправно-

стей. 3) Моделирование неисправностей функциональности на заданном тесте.

Группа 2 – реальные эксперименты (на устройстве), без модели функциональности: 4) Синтез теста путем физической эмуляции дефектов в устройстве. 5) Определение списка неисправностей устройства при выполнении диагностического эксперимента. 6) Верификация теста и дефектов в эксперименте на реальном устройстве.

Группа 3 – тестовые эксперименты (верификация), без дефектов: 7) Синтез теста путем сравнения результатов моделирования модели и реального устройства. 8) Синтез функциональности по реальному устройству и заданному тесту. 9) Верификация теста и модели функциональности относительно реального устройства с существующими неисправностями.

Группа 4 – функционированные эксперименты (на рабочих воздействиях), без тестов: 10) Проверка правильности поведения реального устройства на существующих или заданных дефектах. 11) Проверка работоспособности устройства относительно существующей модели в процессе функционирования. 12) Верификация модели функциональности и списка дефектов относительно поведения реального устройства.

Наиболее популярными задачами из списка перечисленных выше являются: 1, 3, 5, 8, 9. Можно ввести и другую классификацию типов задач, которая дает возможность определить на графе $G = (F, U, T, L)$ все концептуальные пути решения целевых проблем: синтеза тестов, определения модели функциональности, генерирования модели дефектов и проектирования устройства:

- 1) $T = F \oplus L$; 4) $F = T \oplus L$; 7) $L = T \oplus F$; 10) $U = T \oplus L$;
 2) $T = U \oplus L$; 5) $F = U \oplus L$; 8) $L = T \oplus U$; 11) $U = T \oplus F$;
 3) $T = F \oplus U$; 6) $F = T \oplus U$; 9) $L = F \oplus U$; 12) $U = F \oplus L$.

Все конструкции, представленные в отношениях, обладают замечательным свойством обратимости. Любой компонент, вычисляемый через два других, может быть использован в качестве аргумента для определения любого из двух исходных. Потому здесь можно говорить о транзитивной обратимости каждой триады отношений на полном графе, когда по двум любым компонентам всегда и однозначно можно восстановить или определить третий. При этом следует иметь в виду, что формат представления каждого компонента должен быть одинаковым по форме и размерности (векторы, матрицы).

Далее рассмотрены более подробно отдельные задачи синтеза тестов и диагностирования дефектов (функциональных нарушений).

2. Синтез тестов $F \oplus L$ -методом

Используется уравнение пространства $f(F, T, L) = 0 \rightarrow F \oplus T \oplus L = 0$, которое трансформи-

руется к виду $T = F \oplus L$. Например, для функциональности логического og-элемента синтез теста можно представить в виде T-процедуры:

$$T = (F \oplus L) \cap F,$$

где L есть матрица одиночных дефектов переменных логического элемента, задающая возможность транспортирования дефектов от входных линий к выходу, а F – таблица истинности:

$$F = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array}; L = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ 0 & 1 & 1 \\ \hline \end{array}$$

Выполнение T-процедуры связано с применением хог-операции между векторами F и L. Количество векторных операций равно произведению мощностей векторов в каждой таблице:

$$Q(T) = (\text{card}F \times \text{card}L) \times \text{card}F = \text{card}F^2 \times \text{card}L.$$

Для og-элемента T-процедура имеет следующий вид:

$$\begin{array}{l} \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ \hline \end{array} \rightarrow T_i \in F \rightarrow \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ 0 & 0 & 0 \\ \hline \end{array} \\ \\ \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ \hline \end{array} \rightarrow T_i \in F \rightarrow \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 1 & 1 \\ 0 & 0 & 0 \\ \hline \end{array} \end{array} \left. \vphantom{\begin{array}{l} \\ \\ \end{array}} \right\} \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ \hline \end{array}$$

Если функциональность задана в виде кубического покрытия в троичном алфавите, то для синтеза тестов необходимо использовать несимметричную хог-операцию, которая имеет вид:

$$\begin{array}{|c|c|} \hline \tilde{\oplus} & 0 \ 1 \\ \hline 0 & 0 \ 1 \\ 1 & 1 \ 0 \\ x & 1 \ 0 \\ \hline \end{array}$$

Для кубического покрытия and-элемента T-процедура $T = (F \tilde{\oplus} L) \cap F$ имеет следующий вид:

$$\begin{array}{l} \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 0 \\ X & 0 & 0 \\ 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ \hline \end{array} \rightarrow T_i \in F \rightarrow \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 0 & 1 & 0 \\ \hline \end{array} \\ \\ \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 0 \\ X & 0 & 0 \\ 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ \hline \end{array} \rightarrow T_i \in F \rightarrow \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 1 & 0 & 0 \\ \hline \end{array} \end{array} \left. \vphantom{\begin{array}{l} \\ \\ \end{array}} \right\} \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline \end{array}$$

В общем случае выполнение T-процедуры для произвольной функциональности, представленной кубическим покрытием, показано в следующем примере:

$$\begin{array}{c}
 \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline X & 0 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 \\ \hline X & 0 & 1 & 0 \\ \hline X & 1 & 0 & 0 \\ \hline 0 & 1 & X & 0 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 \\ \hline \end{array} \\
 \\
 = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 \\ \hline \end{array}
 \end{array}$$

После синтеза тестов необходимо использовать тест и матрицу дефектов для построения компактной таблицы неисправностей $\bar{F} = (T \oplus L)$, которая предназначена для выполнения процедуры поиска дефектов:

$$\begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ \hline 1 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ \hline \end{array}$$

В данном случае полученная таблица $\bar{F} = (T \oplus L)$ есть подмножество функциональности $\bar{F} \in F = (T \oplus L)$, отдельные координаты которой отмечены символом инверсии, что означает проверяемость неисправности, инверсной по отношению к исправному состоянию. Формально синтез таблицы неисправностей использует некоммутативную операцию сжатия двух битов данных, представленную в следующем виде:

$$\begin{array}{|c|c|} \hline \oplus & 0 & 1 \\ \hline 0 & 0 & \bar{0} \\ \hline 1 & 1 & \bar{1} \\ \hline \end{array}$$

Ниже предлагается вариант синтеза таблицы неисправностей для более сложной функциональности (точками обозначены нули):

$$\begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 1 & . & . & 1 \\ \hline 1 & 1 & 1 & 1 \\ \hline . & 1 & 1 & 1 \\ \hline . & 1 & . & 1 \\ \hline . & 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 1 & 1 & \bar{0} \\ \hline 1 & 1 & 1 & \bar{1} \\ \hline 1 & 1 & 0 & \bar{0} \\ \hline 1 & 0 & 1 & \bar{0} \\ \hline 1 & 0 & 0 & \bar{1} \\ \hline \end{array}$$

Компактная таблица неисправностей является более предпочтительной для визуального оценивания результатов взаимодействия теста и дефектов, проверяемых на каждом входном наборе. При этом выигрыш в объеме памяти для хранения таблицы отсутствует.

3. Аппарат булевых производных для синтеза тестов

Рассматриваются методы взятия булевых производных по таблице истинности, дизъюнктивной форме или кубическому покрытию для создания условий активизации входных переменных при синтезе тестов. Исследование метода предлагается выполнить с помощью следующих трех примеров логических функций:

$$1) f(x) = x_1 \vee x_1 \bar{x}_2. \quad 2) f(x) = x_1 x_2 \vee \bar{x}_1 x_3. \quad 3) f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3.$$

Вопросы, подлежащие решению: 1) Определение всех производных первого порядка по аналитической, кубической и табличной форме задания логической функции. 2) Верификация полученных условий активизации путем их моделирования на одной из форм описания функциональности. 3) Синтез тестов активизации переменных логической функции на основе вычисления производных.

Пример 1. Определить все производные первого порядка по аналитической форме логической функции $f(x) = x_1 \vee x_1 \bar{x}_2$.

Применение формулы вычисления

$$f'(x_i) = \frac{df(x_1, x_2, \dots, x_i, \dots, x_n)}{dx_i} = f(x_1, x_2, \dots, x_i = 0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_i = 1, \dots, x_n)$$

определяет булеву производную первого порядка как сумму по модулю два нулевой и единичной остаточных функций.

Для рассматриваемой функции получается:

$$\begin{aligned}
 \frac{df(x_1, x_2)}{dx_1} &= f(0, x_2) \oplus f(1, x_2) = \\
 &= (0 \vee 0 \bar{x}_2) \oplus (1 \vee 1 \bar{x}_2) = 0 \oplus 1 = 1;
 \end{aligned}$$

$$\begin{aligned}
 \frac{df(x_1, x_2)}{dx_2} &= f(x_1, 0) \oplus f(x_1, 1) = \\
 &= (x_1 \vee x_1 \cdot \bar{0}) \oplus (x_1 \vee x_1 \cdot \bar{1}) = (x_1 \vee x_1 \cdot 1) \oplus (x_1 \vee x_1 \cdot 0) = \\
 &= (x_1 \vee x_1) \oplus (x_1 \vee 0) = x_1 \oplus x_1 = 0.
 \end{aligned}$$

Нулевое значение производной означает отсутствие условий активизации переменной x_2 , что дает основания считать ее несущественной, а следовательно, убрать из числа переменных, формирующих функциональность.

Пример 2. Определить все производные первого порядка по аналитической форме логической функции $f(x) = x_1 x_2 \vee \bar{x}_1 x_3$. Для рассматриваемой функции выполняются следующие вычисления:

$$\begin{aligned}
 \frac{df(x_1, x_2, x_3)}{dx_1} &= f(0, x_2, x_3) \oplus f(1, x_2, x_3) = \\
 &= (0 \cdot x_2 \vee \bar{0} \cdot x_3) \oplus (1 \cdot x_2 \vee \bar{1} \cdot x_3) = \\
 &= (0 \vee 1 \cdot x_3) \oplus (x_2 \vee 0 \cdot x_3) = \\
 &= x_3 \oplus x_2 = x_2 \bar{x}_3 \vee \bar{x}_2 x_3;
 \end{aligned}$$

$$\frac{df(x_1, x_2, x_3)}{dx_2} = f(x_1, 0, x_3) \oplus f(x_1, 1, x_3) = \bar{x}_1 x_3 \oplus (x_1 \vee x_3) =$$

$$= \overline{\bar{x}_1 x_3 (x_1 \vee x_3) \vee \bar{x}_1 x_3 (\bar{x}_1 \vee x_3)} = (x_1 \vee \bar{x}_3)(x_1 \vee x_3) \vee \bar{x}_1 x_3 \bar{x}_1 \bar{x}_3 = x_1.$$

$$\frac{df(x_1, x_2, x_3)}{dx_3} = f(x_1, x_2, 0) \oplus f(x_1, x_2, 1) = x_1 x_2 \oplus (\bar{x}_1 \vee x_2) =$$

$$= \overline{x_1 x_2 (\bar{x}_1 \vee x_2) \vee x_1 x_2 (\bar{x}_1 \vee x_2)} = (\bar{x}_1 \vee \bar{x}_2) (\bar{x}_1 \vee x_2) \vee x_1 x_2 x_1 \bar{x}_2 = \bar{x}_1.$$

Для трех переменных получены 4 условия активизации, которые соответствуют четырем логическим путям в схемной структуре дизъюнктивной формы данной функции.

Пример 3. Определить все производные первого порядка по кубической форме логической функции:

$$f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3 =$$

x ₁	x ₂	x ₃	Y
X	0	0	1
1	1	1	1
X	0	1	0
X	1	0	0
0	1	X	0

$$=$$

x ₂ x ₃	00	01	11	10
x ₁	0	1	0	0
	0	1	0	1

$$=$$

x ₁	x ₂	x ₃	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Процесс-модель вычисления производной по переменной x_i для функции, заданной табличной формой, имеет следующие пункты. 1) Моделирование по таблице истинности (кубическому покрытию) входных наборов для определения столбца Y_i^0 , где переменная x_i имеет только нулевое значение для всех строк таблицы истинности. Число таких наборов всегда равно $q = 2^{n-1}$, n – число переменных. 2) Вычисление координат столбца Y_i^1 с единичным значением переменной x_i для всех строк таблицы. 3) Вычисление столбца $Y_i^\oplus = Y_i^0 \oplus Y_i^1$ с учетом правила $0 \oplus X \vee 1 \oplus X = X$. 4) Формирование дизъюнктивной формы производной функции по единичным значениям столбца Y_i^\oplus без переменной x_i , по которой берется производная. Иначе, фиксируются строки таблицы, соответствующие единичным значениям столбца Y_i^\oplus , который определяет производную функции. Аналитическая модель процесса взятия производной по функции, представленной таблицей, имеет следующий вид:

$$\frac{df}{dx_i} = f(x_1, x_2, \dots, x_i = 0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_i = 1, \dots, x_n);$$

$$Y_i^\oplus = [Y_i^0 = f(x_1, x_2, \dots, x_i = 0, \dots, x_n)] \oplus [Y_i^1 = f(x_1, x_2, \dots, x_i = 1, \dots, x_n)].$$

Для двух различных табличных форм результат вычисления производной по первой переменной представлен ниже:

$$\frac{df}{dx_1} =$$

x ₁	x ₂	x ₃	Y	Y ₁ ⁰	Y ₁ ¹	Y ₁ [⊕]
X	0	0	1	1	1	0
1	1	1	1	0	1	1
X	0	1	0	0	0	0
X	1	0	0	0	0	0
0	1	X	0	0	X	1

$$= x_2 \vee x_2 x_3 = x_2 x_3;$$

$$\frac{df}{dx_1} =$$

x ₁	x ₂	x ₃	Y	Y ₁ ⁰	Y ₁ ¹	Y ₁ [⊕]
0	0	0	1	1	1	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	1	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	1	0	1	1

$$= x_2 x_3.$$

При неоднозначном значении производной для функции, заданной кубическим покрытием, выбирается терм, имеющий максимальное число переменных. Минимизация производной функции на основе тождества $a \vee ab = a$ не сохраняет условия активизации переменной, по которой берется производная. В самом деле, значение функции от трех переменных при условиях $a\bar{b} \vee ab$ может быть равно нулю (единице), что означает возможность отсутствия изменения функции при активизации переменной «с» (последний столбец карты Карно):

ab	00	01	11	10
c	0	1	0	0
	0	1	0	1

Лемма непересекаемости кубов. Возможность корректного взятия производной для получения теста активизации по переменной x_i ограничивается такой минимальной структурой кубического покрытия или аналитической дизъюнктивной (конъюнктивной) нормальной формой, где пересечение любых кубов (строк таблицы истинности) или термов ДНФ (КНФ) дает пустое множество:

$$\frac{df}{dx_i} = f(x_1, x_2, \dots, x_i = 0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_i = 1, \dots, x_n) \in T \leftrightarrow$$

$$\forall i, j (C_i \cap C_j = \emptyset); i, j = \overline{1, n}; i \neq j.$$

В самом деле, если покрытие, представленное выше, записать по правилам непересекаемости кубов, то все производные будут валидны для синтеза тестов без дополнительной проверки:

$$f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3 =$$

x ₁	x ₂	x ₃	Y
X	0	0	1
1	1	1	1
X	0	1	0
X	1	0	0
0	1	X	0

$$\rightarrow$$

x ₁	x ₂	x ₃	Y
X	0	0	1
1	1	1	1
X	0	1	0
X	1	0	0
0	1	1	0

Чтобы получить такое кубическое покрытие, необходимо выполнять минимизацию всеми существующими методами (карты Карно, Квайна, существенных переменных, неопределенных коэффициентов, бинар-

ного графа) с учетом правила: покрытия нулевых и единичных координат таблицы истинности в процессе минимизации не должны пересекаться. В рассматриваемом случае, когда функциональность переписана с учетом данного правила, даже общее число кубов не изменилось, в то время как покрытие приобрело качество непересекаемости (как у таблицы истинности) для синтеза тестов активизации переменных:

$$\frac{df}{dx_1} = \begin{array}{|c|c|c|c|c|c|c|} \hline x_1 & x_2 & x_3 & Y & Y_2^0 & Y_2^1 & Y_2^\oplus \\ \hline X & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline X & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline X & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline \end{array} = x_2 x_3;$$

$$\frac{df}{dx_1} = \begin{array}{|c|c|c|c|c|c|c|} \hline x_1 & x_2 & x_3 & Y & Y_2^0 & Y_2^1 & Y_2^\oplus \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline \end{array} = x_2 x_3.$$

Вычисление производных по всем входным переменным дает возможность построить тест активизации для функциональности, заданной уже не таблицей истинности, а кубическим покрытием, что может существенно уменьшить время синтеза тестов. Для второй переменной функции $f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3$ процесс вычисления производных для трех различных форм (кубической, табличной и аналитической) имеет следующий вид:

$$\frac{df}{dx_2} = \begin{array}{|c|c|c|c|c|c|c|} \hline x_1 & x_2 & x_3 & Y & Y_2^0 & Y_2^1 & Y_2^\oplus \\ \hline X & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline X & 0 & 1 & 0 & 0 & X & X \\ \hline X & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline \end{array} = \bar{x}_3 \vee x_1 x_3 \vee \bar{x}_3 = \bar{x}_3 \vee x_1 x_3;$$

$$\begin{array}{|c|c|c|c|c|c|c|} \hline x_1 & x_2 & x_3 & Y & Y_2^0 & Y_2^1 & Y_2^\oplus \\ \hline 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline \end{array} = \bar{x}_1 \bar{x}_3 \vee x_1 \bar{x}_3 \vee x_1 x_3 = \bar{x}_3 \vee x_1 x_3;$$

Аналогичный результат получен путем определения производной по дизъюнктивной нормальной форме логической функции:

$$\begin{aligned} \frac{df(x_1, x_2, x_3)}{dx_2} &= f(x_1, 0, x_3) \oplus f(x_1, 1, x_3) = \\ &= (\bar{x}_3) \oplus (x_1 x_3) = (x_3) (x_1 x_3) \vee (\bar{x}_3) (\overline{x_1 x_3}) = \\ &= (x_1 x_3) \vee (\bar{x}_1 \bar{x}_3 \vee \bar{x}_3 \bar{x}_3) = x_1 x_3 \vee \bar{x}_3. \end{aligned}$$

Для третьей переменной функции $f(x) = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3$ производные от трех различных форм (кубической, табличной и аналитической) представлены в следующем виде:

$$\frac{df}{dx_3} = \begin{array}{|c|c|c|c|c|c|c|} \hline x_1 & x_2 & x_3 & Y & Y_3^0 & Y_3^1 & Y_3^\oplus \\ \hline X & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline X & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline X & 1 & 0 & 0 & 0 & X & X \\ \hline 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline \end{array} = \bar{x}_2 \vee x_1 x_2;$$

$$\begin{array}{|c|c|c|c|c|c|c|} \hline x_1 & x_2 & x_3 & Y & Y_3^0 & Y_3^1 & Y_3^\oplus \\ \hline 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline \end{array} = \bar{x}_1 \bar{x}_2 \vee x_1 \bar{x}_2 \vee x_1 x_2 = \bar{x}_2 \vee x_1 x_2;$$

$$\begin{aligned} \frac{df(\bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3)}{dx_3} &= f(x_1, x_2, 0) \oplus f(x_1, x_2, 1) = \\ &= (\bar{x}_2) \oplus (x_1 x_2) = (x_2) (x_1 x_2) \vee (\bar{x}_2) (\overline{x_1 x_2}) = \\ &= (x_1 x_2) \vee (\bar{x}_1 \bar{x}_2 \vee \bar{x}_2 \bar{x}_2) = x_1 x_2 \vee \bar{x}_2. \end{aligned}$$

Таким образом, все результаты по вычислению производных от трех форм задания функции идентичны. Наиболее технологичным является метод взятия производной по таблице истинности. Но использование кубического покрытия имеет меньшую вычислительную сложность в силу компактного представления функциональности за счет введения избыточности (символа X) в двоичный алфавит. Использование аналитической формы предполагает существенное повышение сложности алгоритмов, связанной с применением законов булевой алгебры и минимизации функций, что ограничивает ее применение для решения практических задач.

Процесс-модель получения теста $T = [T_{ij}]$, $i = \bar{1}, k$; $j = \bar{1}, n$ комбинационной функциональности:

- 1) $f'(x_i) = f(x_1, x_2, \dots, x_i = 0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_i = 1, \dots, x_n)$;
- 2) $T = \bigcup_{i=1}^n [f'(x_i) * (x_i = 0) \vee (x_i = 1)]$;
- 3) $T_{ij} = T_{i-1, j} \leftarrow T_{ij} = X$; $T_{1j} = 1 \leftarrow T_{1j} = X$;
- 4) $T = T \setminus T_i \leftarrow T_i = T_{i-r}$, $r = \bar{1}, i-1$, $i = \bar{2}, n$.

1) Вычисление производных по всем n переменным функциональности путем использования одной из форм: аналитической, табличной, кубической. 2) Объединение всех условий (векторов) активизации в таблицу, где каждому вектору путем конкатенации (*) ставится в соответствие изменение переменной, по которой была взята производная, что означает удвоение числа тестовых наборов по отношению к общему количеству (k) условий активизации. 3) Доопределение символа $X = \{0, 1\}$ в координате путем присвоения двоичного значения одноименной координаты в предыдущем векторе для получения теста минимальной длины. 4) Минимизация тестовых векторов путем удаления повторяющихся входных последовательностей.

Рис. 2 иллюстрирует таблицы процесса получения теста в соответствии с пунктами 2-4 алгоритма для функциональности $f = \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3$, представленной схемной структурой.

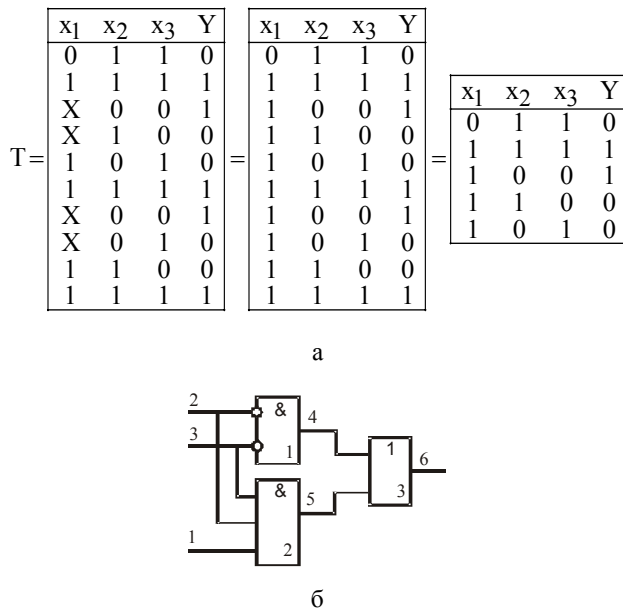


Рис. 2. Таблицы тестов (а) и схемная структура булевой функции (б)

Полученный тест по количеству и качеству идентичен входным наборам, синтезированным ранее с помощью $F \oplus L$ -метода, следовательно, он имеет одинаковые свойства по покрытию неисправностей и глубине поиска дефектов.

Предложенная процесс-модель синтеза тестов для функциональностей может быть использована в качестве встроенного компонента инфраструктуры сервисного обслуживания цифровых систем на кристаллах с использованием стандарта граничного сканирования IEEE 1500.

4. Векторно-логические методы анализа таблиц функциональных нарушений

Аналитическая модель верификации HDL-кода с использованием механизма темпоральных ассерций ориентирована на достижение заданной глубины диагностирования и представлена в следующем виде:

$$\begin{aligned}
 M &= f(F, L, T, C, A, t), \\
 C &= \{C_1, C_2, \dots, C_i, \dots, C_m\}; \\
 L &= \{L_1, L_2, \dots, L_i, \dots, L_n\}; \\
 A(t) &= \{A_1, A_2, \dots, A_i, \dots, A_k\}; \\
 A &\subseteq L; F = L \times C; k \leq n; \\
 T &= \{T_1, T_2, \dots, T_i, \dots, T_p\}.
 \end{aligned}$$

Здесь C_i – группа операторов кода, нагруженная на вершину L_i (переменная, регистр, счетчик, память) и формирующая ее состояние; F – функциональность, представленная транзакционным графом $F = L \times C$ в виде декартова произведения множества вершин и дуг; A – совокупность темпоральных ассерций, как подмножество вершин транзакционного графа $A \subseteq L$. Метод поиска функциональных нарушений (ФН) блока операторов кода использует предварительно построенную таблицу ФН $V = [V_{ij}]$, где строка есть отношение между тестовым сегментом и подмножеством программных блоков $T_i \approx (V_{i1}, V_{i2}, \dots, V_{ij}, \dots, V_{in})$ с возможными ФН. Столбец таблицы формирует отношение между программным блоком и тестовыми сегментами $V_j \approx (T_{1j}, T_{2j}, \dots, T_{ij}, \dots, T_{pj})$, которые могут проверять блок с ФН. На стадии моделирования определяется обобщенная реакция $m = \{m_1, m_2, \dots, m_i, \dots, m_p\}$ механизма ассерций F на тест путем формирования $m_i = (A_1 \vee A_2 \vee \dots \vee A_i \vee \dots \vee A_k)$, $A_i = \{0, 1\}$ как реакции ассерций на тест-сегмент T_i . Поиск ФН основан на определении хог-операции между вектором состояния ассерций и столбцов таблицы ФН $m \oplus (V_1 \vee V_2 \vee \dots \vee V_j \vee \dots \vee V_n)$. Выбор решения определяется совокупностью векторов V_j с минимальным числом единичных координат:

$$V = \min_{j=1, n} [V_j = \sum_{i=1}^p (V_{ij} \oplus m_i)],$$

формирующих программные блоки с ФН, проверяемые на тестовых сегментах.

Повышение адекватности модели диагностирования ФН блоков HDL-кода связано с расширением пространства реакции проекта в процессе его тестовой верификации. В этом случае следует расширить пространство существования механизма ассерций до двумерного путем модификации векторов m и A :

$$\begin{aligned}
 A_i &= (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{ik}); \\
 m_i &= (m_{i1}, m_{i2}, \dots, m_{ij}, \dots, m_{ik}); \\
 V_{ij} &= (V_{ij1}, V_{ij2}, \dots, V_{ijr}, \dots, V_{ijk}); \\
 \{A_{ij}, m_{ij}, V_{ijr}\} &= \{0, 1\}; \\
 i &= \overline{1, p}; \quad j = \overline{1, n}; \quad r = \overline{1, k}.
 \end{aligned}$$

При этом таблица ФН становится трехмерной по параметрам: $i = \overline{1, p}; \quad j = \overline{1, n}; \quad r = \overline{1, k}$ – числа строк или

тест-сегментов, столбцов или функциональных блоков HDL-модели, темпоральных ассерций или точек наблюдения для HDL-кода соответственно:

B	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	m
	B ₁	B ₂	B ₃	B ₄	B ₅	B ₆	B ₇	B ₈	
T ₁	1001	1110	0001	0110	0011	1110	0001	1110	1001
T ₂	1010	1000	0001	0110	0001	0001	1110	1110	1000
T ₃	1110	0001	1111	0111	1011	1110	1001	1101	0011
T ₄	1000	0111	0110	1001	1000	0001	1011	1110	1011
T ₅	0001	0110	1100	0111	0010	1110	0001	1000	1001
T ₆	1110	1011	0110	0001	1110	1100	1001	0111	1110
T ₇	0110	1101	0110	1000	0111	1011	0101	1110	0001
T ₈	1000	1110	1001	0100	0101	1001	0111	0001	1110

Потенциально трехмерность таблицы ФН увеличивает объем диагностической информации, что дает возможность повысить глубину поиска некорректного программного блока, в пределе до оператора или строки кода. Однако для анализа трехмерной таблицы ФН необходимо модифицировать предложенные методы диагностирования.

Поиск дефектов по таблице неисправностей или функциональных нарушений на основе вектора экспериментальной проверки – реакции (m) функциональности на тест $m = (m_1, m_2, \dots, m_i, \dots, m_n), m_i \in \{0, 1\}$ сводится к методам анализа строк или столбцов.

Первый метод основан на применении векторной хог-операции между реакцией m функциональности на тест, формально рассматриваемого в качестве входного вектор-столбца или маски m, и столбцов таблицы неисправностей $m \oplus (B_1 \vee B_2 \vee \dots \vee B_j \vee \dots \vee B_m)$. Для подсчета качества взаимодействия векторов $Q_j(m \oplus B_j)$ в целях выбора лучшего решения определяются столбцы с минимальным числом единиц результирующего вектора

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

Они идентифицируют и формируют дефектные блоки с функциональными нарушениями, проверяемые на тестовых наборах. Аналитическая модель процесса получения решения в виде списка блоков с ФН, присутствующих в программном продукте, представлена в следующем виде:

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

$$B^0 = \bigvee_{j=1, k} [B_j = \sum_{i=1}^n (B_{ij} \oplus m_i) = 0]. \quad (1)$$

Здесь фигурирует вектор экспериментальной проверки, который является входным для последующего анализа таблицы ФН:

$$m = f(A, B) \oplus f^*(A, B, L) \quad (2)$$

есть результат проведения тестового эксперимента – сравнение функционалов (состояний выходов) эталонного $f(A, B)$ и реального $f^*(A, B, L)$ устройства с дефектами L на тестовых наборах A. Во втором случае, если множество $B^0 > 1$, это означает наличие эквивалентных, не различимых на данном тесте и механизме ассерций, функциональных нарушений. Выбор лучшего решения $Y = \bigvee [(Q_1 \wedge Q_2) \oplus Q_1]$ с минимальным числом единичных координат из двух альтернатив, представленных векторными оценками $Q_1 \vee Q_2$, осуществляется с помощью схемы, изображенной на рис. 3.

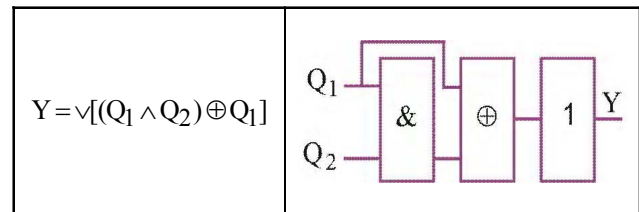
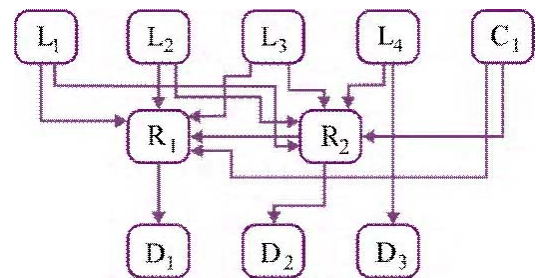


Рис. 3. Схема выбора лучшего решения

Достоинство метода – выбор лучшего решения из всех возможных одиночных и кратных ФН. По существу, в список дефектов включаются такие одиночные ФН, которые при логическом умножении на вектор экспериментальной проверки дают результат в виде соответствующего вектор-столбца. Дизъюнкция всех столбцов, составляющих решение, равна вектору экспериментальной проверки

$\bigvee_{j=1}^r (B_j \in B) = m$. Далее рассматривается пример анализа таблицы ФН блока Row_buffer, представленного на рис. 4.



Test	B ₁	B ₂	B ₃	B ₄	B ₅	B ₆	B ₇	B ₈	B ₉	B ₁₀	m ₁	m ₂
	L ₁	L ₂	L ₃	L ₄	C ₁	R ₁	R ₂	D ₁	D ₂	D ₃		
T ₁	1	0	0	0	0	1	0	1	0	1	0	1
T ₂	0	1	0	0	0	0	1	0	1	0	1	1
T ₃	0	0	1	0	0	1	1	1	0	1	0	1
T ₄	0	0	0	1	0	0	1	0	1	0	1	0
T ₅	0	0	0	0	1	1	1	1	0	1	0	0
T ₆	1	0	0	0	0	0	1	0	1	0	1	1
T ₇	0	1	0	0	0	1	0	1	0	0	0	1
T ₈	0	0	1	0	0	1	0	1	0	0	0	1
T ₉	0	0	1	0	0	0	1	0	1	0	1	0
T ₁₀	0	0	0	1	0	1	1	1	0	0	0	0
T ₁₁	0	0	0	1	0	0	0	0	0	1	0	0
T ₁₂	0	0	0	0	1	0	1	0	1	0	1	0
T ₁₃	0	0	0	0	1	1	1	1	0	0	0	0

Рис. 4. Row_buffer транзакционный граф и таблица ФН

На основе процедуры диагностирования (1) и таблицы ФН (см. рис. 4) можно определить дефектные компоненты методом анализа столбцов таблицы ФН. Здесь векторы m_1, m_2 формируют результаты диагностического эксперимента, выполненные по процедуре (2). Результат диагностирования одиночных и кратных ФН имеет следующий вид:

$$V^s(m_1) = m_1 \wedge \left(\bigvee_{j=1}^{10} B_j \right) = B_9 \rightarrow D_2;$$

$$V^m(m_2) = m_2 \wedge \left(\bigvee_{j=1}^{10} B_j \right) = B_1 \vee B_2 \rightarrow L_1 \vee L_2;$$

$$Q(m_1, D_2) = 1;$$

$$Q[m_2, (L_1 \vee L_2)] = \frac{1}{3} \left(\frac{4}{13} + \frac{1}{4} + 1 \right) = 0,52.$$

В первом случае диагноз определен в виде одного дефектного блока D_2 , присутствующего в транзакционном графе, качество решения равно 1. Во втором случае процедура диагностирования выявила наличие двух дефектных модулей $L_1 \vee L_2$, которые не смогли сформировать идеальную оценку качества. Тем не менее, решение является лучшим среди всех возможных, которое максимально приближено к вектору экспериментальной проверки по критерию принадлежности $Q[m_2, (L_1 \vee L_2)]$. Вычислительная сложность метода анализа столбцов определяется следующей зависимостью:

$$Z^c = 3n^2 + n^2 = 4n^2; Z^r = 3n + n = 4n.$$

Здесь первая оценка учитывает выполнение координатных операций над матрицей, размерностью $n \times n$. Вторая оценка определяет вычислительную сложность регистровых параллельных операций для подсчета критериев качества и обработки матрицы соответственно.

Логический метод анализа строк таблицы неисправностей. Стратегия определения ошибок программного кода по таблице ФН связана с анализом ее строк, состоящим из двух процедур: 1) вычисление логического произведения конъюнкции строк, отмеченных единичными значениями вектора $T_i (m_i = 1)$, на отрицание дизъюнкции нулевых строк $T_i (m_i = 0)$ для одиночных дефектных блоков; 2) вычисление логического произведения дизъюнкции единичных строк на отрицание дизъюнкции нулевых строк для кратных дефектных блоков:

$$V^s = \left(\bigwedge_{m_i=1} T_i \right) \wedge \left(\overline{\bigvee_{m_i=0} T_i} \right);$$

$$V^m = \left(\bigvee_{m_i=1} T_i \right) \wedge \left(\overline{\bigvee_{m_i=0} T_i} \right) \quad (3)$$

Формулы интересны тем, что они не привязаны к критериям качества диагностирования, а оперируют лишь двумя компонентами, таблицей ФН и вектором экспериментальной проверки. Выполнение процедуры диагностирования по формулам (3) для вектора экспериментальной проверки

$$m_1 = (0101010010010),$$

заданного в последней таблице ФН, дает результат:

$V^s(m_1, T) = D_2$, который не хуже, чем ранее полученный методом анализа столбцов. Для вектора экспериментальной проверки $m_2 = (1110011100000)$ резуль-

тат диагностирования имеет вид: $V^m(m_2, T) = L_1 \vee L_2$. Вычислительная сложность метода анализа строк определяется следующей зависимостью: $Z^c = n^2$; $Z^r = n$. Первая оценка предназначена для подсчета числа координатных операций, вторая определяет вычислительную сложность процесса обработки на основе регистровых параллельных операций. Предложенные методы диагностирования ФН для программных и аппаратных продуктов есть один из наиболее существенных компонентов инфраструктуры сервисного обслуживания проектируемых изделий.

Формулы (3) могут быть модифицированы, если ввести следующие обозначения:

$$a = \left(\bigwedge_{m_i=1} T_i \right); b = \left(\bigvee_{m_i=0} T_i \right); c = \left(\overline{\bigvee_{m_i=1} T_i} \right);$$

$$V^s = a\bar{b} = a \oplus ab = a(a \oplus b) = a(b \oplus 1);$$

$$V^m = c\bar{b} = c \oplus cb = c(c \oplus b) = c(b \oplus 1);$$

$$V = \begin{cases} a\bar{b} = a\bar{b} = a \oplus ab = a(a \oplus b) = a(b \oplus 1); \\ c\bar{b} = c\bar{b} = c \oplus cb = c(c \oplus b) = c(b \oplus 1) \leftarrow a\bar{b} = 0 \end{cases}$$

Любое выражение в правой части уравнений может быть использовано для определения функционального нарушения в программном или аппаратном изделии. Различие заключается в наличии или отсутствии инверсии, заменяемой хог-операцией, которая для задач диагностирования и распознавания образов зачастую является более предпочтительной. В таком случае процесс-модель диагностирования одиночных (используется а-компонент) или кратных (b-компонент) дефектов (функциональных нарушений) на основе анализа таблицы ФН будет иметь эффективную векторно-ориентированную вычислительную технологию:

$$V = (b \oplus 1)(a \vee c)$$

встроенного сервисного обслуживания программных и/или аппаратных продуктов. С позиции теории множеств это означает определение результата теоретико-множественного вычитания

$$V = (a \vee c) \setminus b = (a \setminus b) \vee (c \setminus b)$$

в алгебрологическом векторном пространстве.

Для таких операций необходим мультиматричный процессор (ММП), где каждый примитив ориентирован на параллельное выполнение только одной (and, or, xor, slc) операции над соответствующей матрицей (двумерный массив данных). Количество командно-ориентированных матриц-примитивов создает систе-

му – гетерогенный мультиматричный процессор бинарных операций с буфером М (рис. 5).

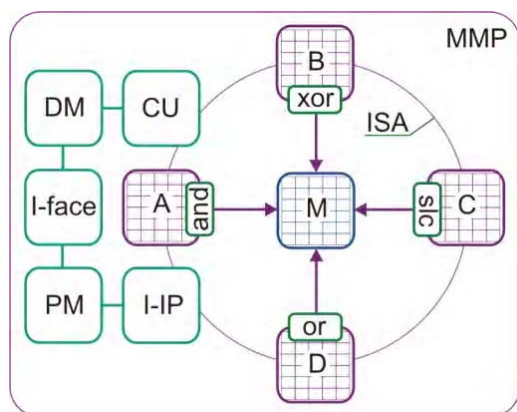


Рис. 5. Мультиматричный процессор бинарных операций

Здесь представлены стандартные блоки: памяти данных DM и программ PM, управления CU, интерфейс I-face и сервисного обслуживания I-IP, а также мультиматричный модуль процессора, включающий 4 блока памяти со встроенными в них операциями (A – and, B – xor, C – or, D – slc – shift left crowding) и буферную память M. Мультиматричный процессор (MMP) ориентирован на параллельное выполнение в данном случае одной из четырех инструкций (ISA – Instruction Set Architecture), оперирующей матрицами двоичных данных одинаковой размерности:

$$M = M \{and, or, xor, slc\} \{A, B, C, D\}$$

с занесением результата в буфер M. Особенность ММП в том, что не ячейка матрицы имеет систему команд из четырех операций, а каждая команда имеет собственную матрицу ячеек в качестве данных для параллельной обработки, что существенно упрощает структуру управления и устройства в целом. Вся сложность ММП перенесена на структуры данных, где память матрицы имеет одну аппаратно-реализованную встроенную команду. Это позволяет иметь примитивную систему управления параллельными вычислительными процессами (SIMD – Single Instruction Multiple Data), последовательностную по своей сути, а значит, нет необходимости создавать сверхсложные компиляторы, ориентированные на распараллеливание вычислительных процессов.

5. Заключение

1. Представлена структурная модель отношений на множестве из четырех основных компонентов технической диагностики (функциональность, устройство, тест, дефекты), которая характеризуется полным хог-взаимодействием всех вершин графа и транзитивной обратимостью каждой триады отношений, что позволяет определить и классифицировать пути решения практических задач, включая синтез тестов, моделирование неисправностей и поиск дефектов.

2. Предложены усовершенствованные методы синтеза тестов для функциональностей, заданных матричными формами описания поведения цифровых компонентов, которые отличаются параллелизмом векторных операций над таблицами, что дает возможность существенно (x2) повысить быстродействие вычислительных процедур.

3. Предложены усовершенствованные методы поиска функциональных нарушений, которые отличаются параллелизмом выполнения векторных операций над строками таблицы ФН, что дает возможность существенно (x2) повысить быстродействие вычислительных процедур, связанных с диагностированием и восстановлением работоспособности программных и аппаратных продуктов.

4. Процесс-модели и методы синтеза тестов для функциональностей и диагностирования нарушений могут быть использованы в качестве встроенных компонентов инфраструктуры сервисного обслуживания цифровых систем на кристаллах с применением стандарта граничного сканирования IEEE 1500.

5. Предложена усовершенствованная процесс-модель определения функциональных нарушений в программном или аппаратном изделии, которая отличается использованием хог-операции, что дает возможность повысить быстродействие диагностирования одиночных или кратных дефектов (функциональных нарушений) на основе параллельного анализа таблицы ФН с помощью логических векторных операций and, or, xor.

Литература: 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, аппаратные 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. 492с. 6. *Хаханов В.И., Хаханова И.В.* VHDL+Verilog = синтез за минуты. Харьков: ХНУРЭ. 2006. 264с. 7. *Хаханов В.И.* Техническая диагностика цифровых и микропроцессорных структур: Учебник. К.: ИСИО, 1995. 242с. 8. *Скобцов Ю.А.* Логическое моделирование и тестирование цифровых устройств / Ю.А. Скобцов, В.Ю. Скобцов. Донецк: ИПММ НАН Украины, ДонНТУ, 2005. 436 с. 9. *IEEE Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture IEEE Std 1149.7-2009.* 985 p. 10. *Da Silva F., McLaurin T., Waayers T.* The Core Test Wrapper Handbook. Rationale and Application of IEEE Std. 1500™. Springer. 2006. XXIX. 276 p. 11. *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. 12. *IEEE Std 1800-2009 IEEE Standard for System Verilog- Unified Hardware Design, Specification, and Verification Language.* <http://ieeexplore.ieee.org/servlet/>

опас?number= 5354133 **13. Marinissen E.J.** Testing TSV-based three-dimensional stacked ICs // DATE 2010. 2010. P.1689-1694. **14. Benso A., Di Carlo S., Prinetto P., Zorian Y.** IEEE Standard 1500 Compliance Verification for Embedded Cores // IEEE Trans. VLSI Syst. 2008. No 16(4). P. 397-407. **15. Ubar R., Kostin S., Raik J.** Embedded diagnosis in digital systems // 26th International Conference "Microelectronics", MIEL 2008. 2008. P. 421-424. **16. Elm M., Wunderlich H.-J.** Scan Chain Organization for Embedded Diagnosis // Design, Automation and Test in Europe, DATE '08. 2008.P. 468-473. **17. Bulent I. Dervisoglu.** A Unified DFT Architecture for Use with IEEE 1149.1 and VSIA/IEEE P1500 Compliant Test Access Controllers. Proceedings of the Design Automation Conference. 2001. P. 53-58. **18. Chenlong Hu, Ping Yang, Ying Xiao, Shaoxiong Zhou.** Hardware design and realization of matrix converter based on DSP & CPLD // 3rd International Conference Power Electronics Systems and Applications. 2009. P. 1-5. **19. Dave N., Fleming K., Myron King, Pellauer M., Vijayaraghavan M.** Hardware Acceleration of Matrix Multiplication on a Xilinx FPGA // 5th IEEE/ACM International Conference Formal Methods and Models for Codesign. 2007. P.97-100. **20. Loucks W.M., Snelgrove M., Zaky S.G.** A Vector Processor Based on One-Bit Microprocessors // IEEE Micro. Volume 2, Issue 1. 1982. P. 53-62. **21. Hilewitz Y., Lauradoux C., Lee R.B.** Bit matrix multiplication in commodity processors // International Conference Application-Specific Systems, Architectures and Processors. 2008. P. 7-12.

Поступила в редколлегию 11.01.2011

Рецензент: д-р техн. наук, проф. Баркалов А.А.

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

Ngene Christopher Umerah, 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.

Литвинова Евгения Ивановна, канд. 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.

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.

Александров Владислав Игоревич, студент гр. КИ-07-8 факультета КИУХНУРЭ. Научные интересы: автоматизация диагностирования и встроенный ремонт компонентов цифровых систем в пакете кристаллов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.