

УДК 519.713:681.326

Ваде Гриби, В.В. Елисеев, Хассан Ктейман, В.И. Хаханов, И.В. Хаханова

## АНАЛИЗ НЕИСПРАВНОСТЕЙ ПРИ ПРОЕКТИРОВАНИИ ДИСКРЕТНЫХ ЛОГИЧЕСКИХ СЕТЕЙ

### 1. Введение

Проблема создания высокопроизводительных мозгоподобных ЭВМ и принципов построения их процессоров в виде логических сетей лежит в основе настоящих исследований. Актуальность работы определяется необходимостью значительного повышения быстродействия средств моделирования и генерации тестов для структурно- и функционально-сложных логических сетей, имплементация которых в цифровые системы на кристаллах, насчитывает миллионы вентилях. Существующие автоматические системы тестирования и моделирования известных фирм: Cadence, Mentor Graphics, Synopsys, Logic Vision способны обрабатывать устройства на кристаллах размерностью порядка миллионов вентилях за несколько часов (временные затраты относятся к рабочим станциям с частотой и оперативной памятью в 0,5 ГГц и 0,5 Гбайт). Тем не менее, предлагаемые модели и алгоритмические решения для их моделирования могут быть конкурентоспособными на рынке тестирования цифровых устройств большой размерности. Они позволяют на порядок повысить быстродействие анализа проекта после выполнения синтеза в целях определения качества тестов проверки неисправностей.

В настоящей статье разрабатываются методы диагностики логических сетей мозгоподобных ЭВМ. Применительно к нейронным сетям мозга диагностика логических сетей соответствует обнаружению дефектов в нейронных сетях и их исправлению.

Объект тестирования — цифровая система, реализующая логическую сеть в кристалле ASIC или PLD, спецификация которой представлена на языке VHDL.

Цель исследования — разработка быстродействующего метода моделирования одиночных константных неисправностей для оценки качества синтезируемых тестов для логических сетей, имплементируемых в кристаллы, содержащие миллионы вентилях.

Задачи исследования: 1. Создание обобщенной модели дедуктивно-параллельного анализа логической цифровой схемы. 2. Разработка дедуктивной модели структурно-функционального анализа цифровых схем в целях поиска множества сходящихся разветвлений (СР) и определения схемных древовидных структур (ДС). 3. Разработка метода

моделирования неисправностей, основанного на раздельной обработке СР и ДС. 4. Алгоритмическая реализация моделирования на основе реконфигурирования модели устройства в процессе анализа неисправностей и применения процедур обратного прослеживания в целях существенного уменьшения времени моделирования тестов и оценки их качества.

Основу ОДП-метода (Backtraced Deductive-Parallel) — обратное дедуктивно-параллельное моделирование неисправностей — составляют: методы повышения быстродействия анализа неисправностей [1–3], дедуктивная модель транспортирования неисправностей [4, 5], параллельный метод обработки списков дефектов функционального элемента [4, 6] и алгоритм обратного прослеживания примитивов [7] при обработке цифрового устройства.

### 2. Обобщенная модель дедуктивно-параллельного анализа неисправностей

Модель дедуктивно-параллельного синхронного анализа неисправностей дискретного объекта позволяет за одну итерацию обработки схемы вычислять все дефекты, проверяемые на двоичном тест-векторе. Такая модель основана на решении следующего уравнения [3,4]:

$$L = T \oplus F, \quad (1)$$

где  $F = (F_{m+1}, F_{m+2}, \dots, F_i, \dots, F_n)$  ( $i = m+1, n$ ) — совокупность функций исправного поведения устройства;  $m$  — число его входов;  $Y_i = F_i(X_{i1}, \dots, X_{ij}, \dots, X_{in_i})$  —  $n_i$ -входовой  $i$ -й элемент схемы, реализующий  $F_i$  для определения состояния линии (выхода)  $Y_i$  на тест-векторе  $T_i$ ; где  $X_{ij}$  —  $j$ -й вход  $i$ -го элемента; двоичный тест  $T = (T_1, T_2, \dots, T_i, \dots, T_k)$  — упорядоченная совокупность двоичных векторов, доопределенная в процессе исправного моделирования на множестве входных, внутренних и выходных линий, где тест-вектор задается в виде

$$T_i = (T_{i1}, T_{i2}, \dots, T_{ij}, \dots, T_{im}), \quad (2)$$

а его необходимая координата вычисляется моделированием функции  $T_{ij} = Y_i = F_i(X_{i1}, \dots, X_{ij}, \dots, X_{in_i})$  на тест-векторе  $T_i$ ;  $L = (L_1, L_2, \dots, L_i, \dots, L_k)$  — множество дедуктивных схем или моделей, определяемых выражением (1), где  $L_i = (L_{i1}, L_{i2}, \dots, L_{ij}, \dots, L_{im})$ ;

$$L_{ij} = T_i \oplus F_i \quad (3)$$

— дедуктивная функция (ДФ) параллельного моделирования неисправностей на тест-векторе  $T_i$ , со-

ответствующая исправному элементу  $F_i$ , которая дает возможность вычислять список входных неисправностей, транспортируемых на выход элемента  $F_i$  [8].

Понятие синхронности введенной модели (1) определяется условием:  $\Delta t = (t_{j+1} - t_j) \gg \tau \gg \tau_i$ , когда интервал времени между сменой входных наборов  $(t_{j+1} - t_j)$ , подаваемых на схему, намного больше максимальной задержки схемы  $\tau$  и элемента  $\tau_i$ . Это позволяет исключить время как несущественный параметр [8], что используется в технологиях моделирования и синтеза тестов.

В общем случае, когда функция устройства представлена таблицей истинности, применение формулы (1) позволяет получить для заданного тест-вектора  $T_i$  таблицу транспортирования неисправностей, по которой можно записать ДФ моделирования дефектов. Пример получения такой функции представлен в следующем виде (первое слагаемое – тест-вектор, второе и результат – таблицы истинности и транспортирования дефектов):

|       |       |       |
|-------|-------|-------|
| $X_1$ | $X_2$ | $Y_1$ |
| 0     | 1     | 0     |

 $\oplus$ 

|       |       |       |
|-------|-------|-------|
| $X_1$ | $X_2$ | $Y_1$ |
| 0     | 0     | 0     |
| 0     | 1     | 0     |
| 1     | 0     | 1     |
| 1     | 1     | 1     |

 $=$ 

|       |       |       |
|-------|-------|-------|
| $X_1$ | $X_2$ | $L_1$ |
| 0     | 1     | 0     |
| 0     | 0     | 0     |
| 1     | 1     | 1     |
| 1     | 0     | 1     |

Здесь дедуктивная функция  $L_1 = X_1 X_2 \vee X_1 \bar{X}_2$  записана в виде дизъюнктивной нормальной формы по конstituентам единицы таблицы транспортирования дефектов.

С учетом разбиения теста на составляющие векторы уравнение (1) получения ДФ для  $T_i \in T$  принимает следующий вид:  $L_i = T_i \oplus F_i$ . Если функциональное описание цифрового устройства представлено компонентами (примитивами), формирующими состояния всех линий схемы, то в качестве формулы преобразования исправной модели примитива  $F_i$  на тест-векторе  $T_i$  в дедуктивную функцию  $L_i$  выступает выражение

$$L_i = T_i \oplus F_i = f_{ii}[(X_{i1} \oplus T_{i1}), (X_{i2} \oplus T_{i2}), \dots, (X_{iy} \oplus T_{iy}), \dots, (X_{in_i} \oplus T_{in_i})] \oplus T_{ii}, \quad (4)$$

которое является основой дедуктивного анализа цифровых проектов [3, 6].

**Пример 1.** Получить дедуктивные функции параллельного моделирования неисправностей на исчерпывающем тесте для базиса логических элементов И, ИЛИ, НЕ.

Используя выражение (4), выполним следующие очевидные преобразования для логического элемента И:

$$\begin{aligned} L\{T = (00, 01, 10, 11), F = (X_1 \wedge X_2)\} &= \\ &= L\{(\bar{x}_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee x_1 \bar{x}_2 \vee x_1 x_2) \wedge \\ &\wedge [(X_1 \oplus T_{i1} \wedge X_2 \oplus T_{i2}) \oplus T_{i3}]\} = \\ &= (\bar{x}_1 \bar{x}_2)\{[(X_1 \oplus 0) \wedge (X_2 \oplus 0)] \oplus 0\} \vee \\ &\vee (\bar{x}_1 x_2)\{[(X_1 \oplus 0) \wedge (X_2 \oplus 1)] \oplus 0\} \vee \\ &\vee (x_1 \bar{x}_2)\{[(X_1 \oplus 1) \wedge (X_2 \oplus 0)] \oplus 0\} \vee \\ &\vee (x_1 x_2)\{[(X_1 \oplus 1) \wedge (X_2 \oplus 1)] \oplus 1\} = \\ &= (\bar{x}_1 \bar{x}_2)(X_1 \wedge X_2) \vee (\bar{x}_1 x_2)(X_1 \wedge \bar{X}_2) \vee \\ &\vee (x_1 \bar{x}_2)(\bar{X}_1 \wedge X_2) \vee (x_1 x_2)(X_1 \vee X_2); \end{aligned}$$

Аналогичные преобразования для логического элемента ИЛИ дают следующий результат:

$$\begin{aligned} L\{T = (00, 01, 10, 11), F = (X_1 \vee X_2)\} &= \\ &= (\bar{x}_1 \bar{x}_2)(X_1 \vee X_2) \vee (\bar{x}_1 x_2)(\bar{X}_1 \wedge X_2) \vee \\ &\vee (x_1 \bar{x}_2)(X_1 \wedge \bar{X}_2) \vee (x_1 x_2)(X_1 \wedge X_2). \end{aligned}$$

Здесь  $T_i = (T_{i1}, T_{i2}, T_{i3}), (i = \bar{1}, 4)$  – тест-вектор, имеющий 3 координаты, где последняя определяет состояние выхода двухвходового элемента И (ИЛИ).

В следующем преобразовании для инвертора тест-вектор имеет 2 координаты:  $T_i = (T_{i1}, T_{i2}), (i = \bar{1}, 2)$ , где последняя – состояние выхода элемента:

$$\begin{aligned} L\{T = (0, 1), F = \bar{X}_1\} &= L\{(\bar{x}_1 \vee x_1)[(\bar{X}_1 \oplus T_{i1}) \oplus T_{i2}]\} = \\ &= \bar{x}_1 \wedge [(X_1 \oplus 0) \oplus 1] \vee x_1 [(X_1 \oplus 1) \oplus 0] = \\ &= \bar{x}_1 \bar{X}_1 \vee x_1 \bar{X}_1 = \bar{x}_1 X_1 \vee x_1 \bar{X}_1. \end{aligned}$$

Последнее выражение иллюстрирует инвариантность операции инверсии ко входному набору для транспортирования дефектов. Она трансформируется в повторитель. Поэтому данная функция не фигурирует на выходах дедуктивных элементов. Совместная аппаратная реализация ДФ для двухвходовых элементов И, ИЛИ на исчерпывающем тесте представлена универсальным функциональным примитивом (рис. 1) дедуктивно-параллельного анализа неисправностей.

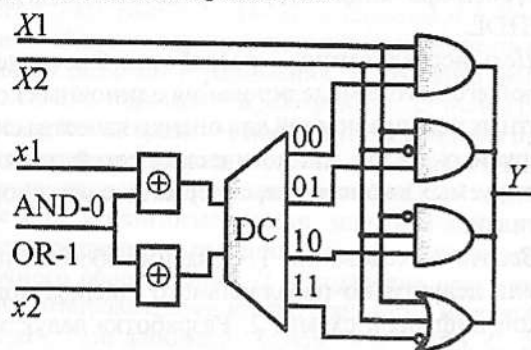


Рис. 1. Симулятор неисправностей

В симуляторе представлены булевы ( $x_1, x_2$ ) и регистровые ( $X_1, X_2$ ) переменные, сигнал выбора

типа исправной функции (AND, OR), выходная регистровая переменная  $Y$ . Состояния двоичных входов  $x_1, x_2$  и AND/OR выбирают одну из четырех дедуктивных функций для получения вектора  $Y$  проверяемых неисправностей. Для иллюстрации параллельного моделирования входных 4-разрядных векторов неисправностей в целях получения на выходе  $Y$  множества проверяемых дефектов для логических элементов 2И, 2ИЛИ используется следующая таблица:

| And/Or | $x_1 x_2$ | $X_1$ | $X_2$ | $Y$  |
|--------|-----------|-------|-------|------|
| 0      | 00        | 0111  | 1011  | 0011 |
| 1      | 00        | 0111  | 1011  | 1111 |
| 0      | 11        | 0101  | 0110  | 0111 |
| 1      | 11        | 1101  | 0111  | 0101 |

Применение такого симулятора дает возможность трансформировать вентильную модель  $F$  исправного поведения схемы в дедуктивную  $L$ , которая инвариантна в смысле универсальности тестовым наборам и не предполагает в процессе моделирования использовать модель  $F$ . Поэтому симулятор, как аппаратная модель ДФ, ориентирован на создание встроенных средств дедуктивно-параллельного моделирования, повышающих быстродействие анализа в 10–1000 раз по сравнению с программной реализацией. Но при этом соотношение объемов вентильных (после синтеза) моделей исправного моделирования и анализа неисправностей составляет 1:10. Подход аппаратного анализа неисправностей направлен на расширение функциональных возможностей встроенных средств исправного моделирования (HEST<sup>TM</sup> – Hardware Embedded Simulator) фирмы Aldec [9]. Вычислительная сложность обработки цифрового устройства, состоящего из  $n$  вентилях, равна  $Q = (2n^2\tau)/W$ , где  $\tau$  – время выполнения регистровой операции (AND, OR, NOT);  $W$  – разрядность регистра.

### 3. Дедуктивный метод структурного анализа

Метод необходим для определения сходящихся разветвлений в структуре цифрового устройства.

Имеется вершина  $V_j \in V$  в ориентированном графе  $G$  линий схемы. По отношению к смежным вершинам-предшественникам, которые являются прообразами  $f^{-1}(V_j)$ , упомянутая вершина  $V_j \in V$  является образом (рис. 2), где  $f^{-1}(\dots)$ , – оператор взятия прообраза.

**Определение 1.** Каждая вершина бесконтурного графа имеет единственное объединение ее образа и суперпозиции прообразов:

$$f^*(V_j) = f^{-1}(f^{-1}(\dots(f^{-1}(V_j)))) \cup V_j, \quad (5)$$

которое называется экстраобразом. Иначе, экстраобраз есть множество всех предшественников для  $V_j$ , включая и вершину  $V_j$ .

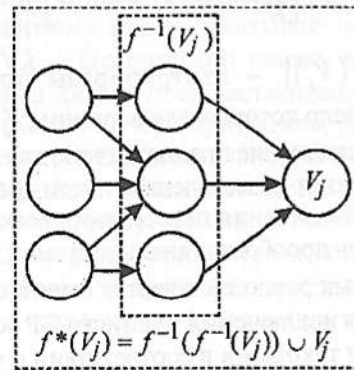


Рис. 2. Смежность как соответствие прообразов образу

**Утверждение 1.** Все вершины (линии), входящие в контур, имеют одинаковые экстраобразы.

Это является следствием таких фактов: 1) все линии, принадлежащие контуру, достижимы по отношению друг к другу; 2) каждая линия, входящая в контур, имеет в качестве предшественника любую вершину, от которой существует путь к одной из контурных линий. Отсюда следует, что все вершины  $V_j \in V$ , принадлежащие контуру, имеют одинаковые экстраобразы [10].

Сходящемуся разветвлению могут предшествовать линии, которые не являются таковыми, но проявляются на вершине-схождении как СР. Пример таких линий есть вершины, составляющие путь без ветвлений, заканчивающийся на линии СР.

**Утверждение 2.** Для того чтобы устранить всех предшественников сходящегося разветвления, входящих в множество  $V^j$  кандидатов в СР для вершины  $V_j$ , но не являющихся СР, необходимо и достаточно вычесть из множества  $V^j$  объединение пересечений всех пар сочетаний  $C_{n_j}^2$  экстраобразов  $f^*(f^{-1}(V_j))$  для непосредственных предшественников вершины  $V_j$ , где  $n_j$  – число предшественников.

**Процедура дедуктивного анализа.** Если граф – ориентированный, без контуров глобальных обратных связей, то стратегия поиска сходящихся разветвлений сводится к одноразовому проходу всех его вершин на основе процедуры дедуктивной обработки  $V_j \in V$ , включающей операции:

$$1) V^j = \bigcup_{C_{n_j}^2} \left[ f^*(f_p^{-1}(V_j)) \bigcap_{i=1, n_j-1}^{p=i, q=i+1, n_j} f^*(f_q^{-1}(V_j)) \right];$$

$$2) V^j = V^j \setminus \bigcup_{C_{m_j}^2} \left[ f^*(V_p^j) \bigcap_{i=1, m_j-1}^{p=i, q=i+1, m_j} f^*(V_q^j) \right];$$

$$3) V^{RC} = V^{RC} \cup V^j; \quad (6)$$

$$4) V_j = \left[ \bigcup_{i=1}^{n_j} f^*(f_i^{-1}(V_j)) \cup V_j \right] \setminus V^{RC}.$$

где  $f^*(f_i^{-1}(V_j))$  — экстраобразы прообразов  $f_i^{-1}(V_j)$ , число которых для вершины  $V_j$  равно  $n_j$ .

Первое уравнение предназначено для определения сходящихся разветвлений путем выполнения операции объединения пересечений всех пар экстраобразов для прообразов анализируемой вершины, число которых равно количеству сочетаний —  $C_{n_j}^2$ . Второе — для исключения из списка СР вершин, не являющихся таковыми в соответствии с утверждением 2. Здесь  $f^*(V_j^i)$  — экстраобраз вершины, включенной в список кандидатов в СР при анализе  $V_j \in V$ , число которых равно  $m_j$ . Третье уравнение предназначено для накопления множества СР в целях его последующего вычитания из списков предшественников для каждой вершины. Оно необходимо для уменьшения мощности прообразов обрабатываемой вершины, которое будет существенней при больших коэффициентах разветвлений у линий предшественников, уже отнесенных к списку  $V^{RC}$ . Четвертое уравнение предназначено для формирования экстраобраза обрабатываемой вершины  $V_j \in V$ .

Время, затрачиваемое на поиск СР при работе описанного препроцессора (6), практически не отличается от времени моделирования дефектов на входном наборе дедуктивным или параллельным методом.

**Определение 2.** Линия, принадлежащая контуру обратной связи, является СР, если она имеет более одной исходящей дуги, а также схождение на линии, не принадлежащей данному контуру.

Если в схеме присутствуют глобальные обратные связи, то перед выполнением процедуры поиска СР необходимо сделать их псевдоразрыв, например по алгоритму, приведенному в [8], а затем применить описанные выше модели идентификации СР.

#### 4. Топологический ОДП-метод моделирования

Использованию процедуры обратного прослеживания [7] в общем случае препятствует невозможность выполнения одномерной активизации дефектов, которая может быть проиллюстрирована двумя вариантами некорректности, представленными следующим примером.

**Пример 2.** Выполнить анализ схем (рис. 3) в целях определения множества проверяемых дефектов на заданных тест-векторах с помощью процедуры обратного прослеживания.

В левой схеме константная неисправность  $2^1$  ложно проверяется на одномерном пути 2-5-6 с по-

мощью процедуры обратного прослеживания. Однако при этом не учитывается тот факт, что данный дефект изменяет состояние линии 4 с 1 на 0, что создает условия запрета транспортирования неисправности на выход схемы. Поэтому неисправность на линии  $a - 2^1$  является ложнопроверяемой на наборе (101) при рассмотрении ее транспортирования по одномерному пути активизации.

Правая схема является примером альтернативной ситуации — к выходу 6 нет одномерного пути транспортирования дефектов с внешних входов и, в частности, от линии  $b$ . Тем не менее, неисправность  $2^1$  на упомянутой линии проверяется, поскольку ее присутствие создает кратный дефект на линиях 4 и 5, который изменяет состояние выходной линии схемы.

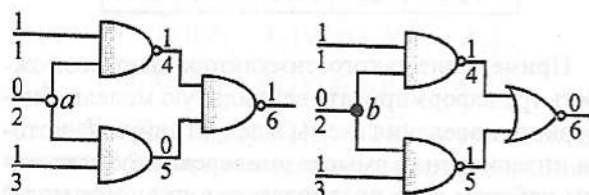


Рис. 3. Ложная проверка и не проверка дефектов

В обоих случаях имеется существенная некорректность, которая не позволяет использовать идею одномерности при отслеживании пути транспортирования неисправностей от выходов ко входам схемы.

Учитывая, что некорректность связана исключительно с одномерной активизацией неисправностей сходящихся разветвлений, далее предлагается сначала выполнить их моделирование, а затем исключить из рассмотрения, сделав структуру схемы древовидной и пригодной для одномерного обратного прослеживания.

Поскольку ниже рассматривается анализ неисправностей в привязке к топологии схемы цифрового устройства, то естественным представляется назвать метод анализа дедуктивно-топологическим (ДТ) на основе обратного прослеживания. Основные пункты стратегии моделирования (см. рис. 4) можно представить шагами:

1. Препроцессор структурного анализа модели схемы — осуществляет поиск СР в схеме цифрового устройства. Вычислительная сложность данной процедуры  $Q_r = n^2$ , но она является разовой и поэтому практически не влияет на быстродействие моделирования входных наборов в целом.

2. Фаза исправного моделирования цифровой схемы. Предназначена для определения реакции всех невыходных линий устройства на входной набор  $T_i \in T = [T_n]$ .

3. Преобразование схемы на текущем тест-векторе в дедуктивную модель по правилам (4).

4. Моделирование неисправностей сходящихся разветвлений по дедуктивной модели схемы, количество которых значительно меньше остальных линий;

5. Обратное одномерное прослеживание проверяемых неисправностей древовидных подграфов по дедуктивной модели устройства.

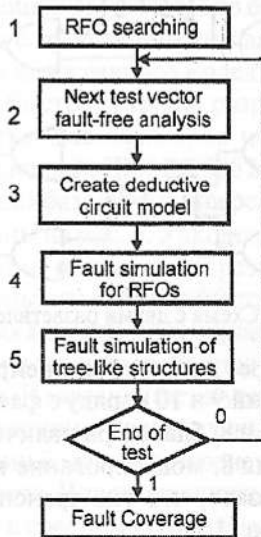


Рис. 4. Стратегия ДТ-моделирования

Теоретическим обоснованием применения данного пункта является доказательство следующих теорем и формулировка следствий из них.

**Лемма.** Сходящиеся разветвления в комбинационной схеме являются причиной появления кратных дефектов на входах примитивов.

**Доказательство.** Пусть комбинационная схема не имеет сходящихся разветвлений. В этом случае она представлена древовидной структурой. Тогда пересечение подграфов-предшественников (экстраобразов) для любых входов примитивного элемента определяется как

$$f^*(X_i) \cap f^*(X_j) = \emptyset. \quad (7)$$

Это справедливо для примитивов первого уровня, где входы элементов являются входами схемы, занумерованными различными идентификаторами линий. Если примитив находится внутри схемы, то в силу древовидности (7) любые два входа не будут иметь хотя бы одного общего предшественника. Иначе нарушится условие древовидности графовой структуры. Таким образом, два любых входа примитива древовидной схемы, не имея общих предшественников, не будут иметь и общих неисправностей на конкретном двоичном входном наборе, которые могут быть транспортированы через рассматриваемый элемент.

**Теорема 1.** Для древовидной структуры дедуктивной схемы инверсный вход примитива И:  $\bar{X}_i \in L$  запрещает транспортирование всех неисправностей, принадлежащих линиям-предшественникам.

**Доказательство.** Во-первых, инверсия на входе может иметь место на примитиве, имеющем две и более входные линии, во-вторых, она означает вычитание списка неисправностей, принадлежащего данному входу, учитывая изоморфизм  $\bar{X}_i X_j \approx X_j \setminus X_i$ . Но согласно лемме, пересечение двух списков линий-предшественников, относящихся ко входам одного примитива, равно пустому множеству (7). Следовательно, вычитание списка линий-предшественников никогда не уменьшит мощность любого другого списка, относящегося к неинверсному входу рассматриваемого элемента. Таким образом, для древовидных структур комбинационных схем всегда будет выполняться условие:

$$f^*(X_i) \setminus f^*(X_j) = f^*(X_i) \leftarrow f^*(X_i) \cap f^*(X_j) = \emptyset.$$

**Теорема 2.** Если в древовидной структуре дедуктивной схемы L существует примитив И, имеющий более одного неинверсного входа  $X_i X_j$ , то такой элемент запрещает транспортирование всех неисправностей, принадлежащих линиям-предшественникам всех его входов.

**Доказательство.** Наличие двух и более неинверсных, прямых входов в дедуктивном элементе означает пересечение списков линий, являющихся предшественниками для рассматриваемых входов. Но согласно лемме, входы одного примитива ДС не имеют общих предшественников, отсюда следует, что пересечение упомянутых списков будет всегда равно пустому множеству:

$$f^*(X_i) \cap f^*(X_j) = \emptyset \leftarrow X_i X_j.$$

Если же дедуктивный элемент имеет входы с инверсией, то по теореме 1 вычитание из полученного пустого множества предшественников любого непустого, принадлежащего входу с инверсией, дает также пустой результат.

Следовательно, дедуктивный элемент И, имеющий более одного неинверсного входа, не пропустит через себя неисправности линий предшественников.

**Следствия:** 1) Инверсный вход дедуктивного элемента И запрещает активизацию всех неисправностей линий-предшественников, относящихся к данному входу. 2) Если дедуктивный примитив имеет более одного неинверсного входа, то все его входы следует определить инверсными. 3) Дедуктивный элемент И пропустит неисправности линий-предшественников только по неинверсному входу, который должен быть единственным. 4) Дедуктивный элемент ИЛИ не может иметь инверсных входов. 5) Сходящееся разветвление отмечается инверсией (кружком на линии ветвления), если его неисправность не проверяется на тест-векторе. 6) Инверсия на линии, входной или выходной, является условием разрыва активизации неисправ-

ностей и прекращения обратного прослеживания дефектов по рассматриваемой ветви древовидной структуры. 7) Проверяемую на тест-векторе линию сходящегося разветвления (на топологическом рисунке схемы далее отмечается жирным кружочком) при выполнении процедуры обратного прослеживания следует рассматривать как наблюдаемый выход схемы. 8) Интерпретация результата топологического моделирования: неисправности линий, не отмеченные знаками инверсий на дедуктивной модели схемы, проверяются.

В качестве иллюстрации основных шагов топологического моделирования ниже предлагаются два примера.

**Пример 3.** Пусть дано цифровое устройство (рис. 5, первая схема, при рассмотрении сверху вниз), содержащее 3 сходящихся разветвления. Входной вектор: 1011111. Необходимо определить проверяемые дефекты константного типа. Состояния линий после исправного моделирования представлены в скобках. Результат процесса преобразования исправной модели устройства в дедуктивную и моделирование неисправностей линий СР зафиксирован на второй схеме. Все неисправности разветвлений (2, 10, 13), инверсные по отношению к исправному состоянию этих линий, являются проверяемыми. Этот факт отмечен на схеме черными кружочками.

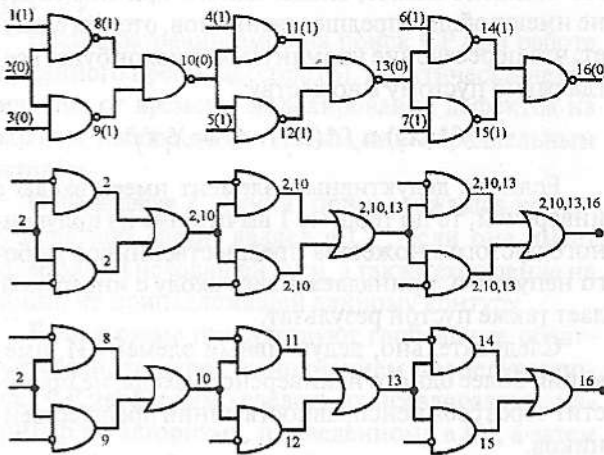


Рис. 5. Моделирование схемы с разветвлениями

Процедура обратного прослеживания неисправностей в целях определения списка проверяемых заключается в построении максимальных древовидных подграфов, ограниченных на топологии прозрачными кружочками. При этом черные кружочки есть проверяемые линии разветвления, которые следует рассматривать на третьей схеме (см. рис. 5) как наблюдаемые выходы. Здесь же обозначены все неисправности, проверяемые на тест-векторе: {2, 8, 9, 10, 11, 12, 13, 14, 15, 16}. Иллюстрация результата моделирования: неисправности линий, не отмеченные прозрачными кружочками (знаками инверсии), проверяются.

**Пример 4.** Определить список проверяемых на тест-векторе 101111 неисправностей для цифрового устройства, представленного на рис. 6, содержащего 2 сходящихся разветвления.

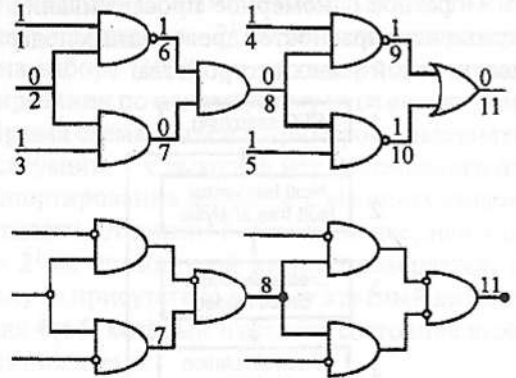


Рис. 6. Схема с двумя разветвлениями

Здесь интерес представляет непроверка неисправностей линий 9 и 10 наряду с фактом проверяемости линий 7 и 8, благодаря наличию сходящегося разветвления 8, моделирование неисправности которого показало, что она транспортируется на внешний выход 11.

На рис. 7 представлена структура эволюционного развития и интеграции методов моделирования неисправностей. В основу положены дедуктивный (Д – D) и параллельный (П – P) алгоритмы. Далее был разработан универсальный, но относительно медленный, дедуктивно-параллельный метод (ДП – DP) [11]. Затем был реализован (ОДП – BDP) метод [12], ориентированный на быструю обработку моделей цифровых систем. Последний был модифицирован к ТД (TD)-методу, который ориентирован на вентильный уровень представления цифровых систем. При этом в описании устройства предварительно осуществляется поиск множества сходящихся разветвлений и выделение древовидных структур как дополнение к СР. Для моделирования сходящихся разветвлений (RFO) используется дедуктивно-параллельный алгоритм для анализа древовидных структур (TLS) – ТД-метод. Такое разделение функций обработки цифровых схем большой размерности позволяет как минимум на порядок повысить быстродействие моделирования неисправностей по сравнению с дедуктивным и параллельным методами.

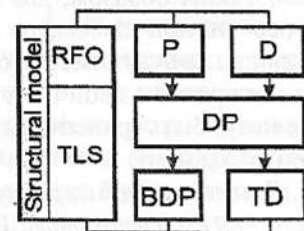


Рис. 7. Эволюция методов моделирования

5. Заключение

Предложенный метод моделирования неисправностей ориентирован на обработку логических сетей, имплементируемых в кристаллы, содержащие миллионы вентилях. Тестовые эксперименты программной реализации метода на сотнях цифровых комбинационных и последовательностных схем дали хорошие результаты по быстродействию по сравнению с традиционными алгоритмами параллельного и дедуктивного моделирования. Примеры анализа быстродействия разработанного метода (обработка тест-примеров на 1000 входных последовательностей, IBM PC 500 МГц, 256 Мбайт) и существующих базовых (университетские реализации) показаны на рис. 8. Ускорение моделирования составляет не менее десяти раз. На рис. 9 представлены результаты анализа быстродействия трех реализованных методов моделирования цифровых схем на одном и том же компьютере при обработке 1000 векторов. Показано преимущество ТД-метода перед дедуктивно-параллельным. Выигрыш в быстродействии более существенен для схем большой размерности. Число сходящихся разветвлений в тест-схемах в среднем составляет 20% от общего количества линий.

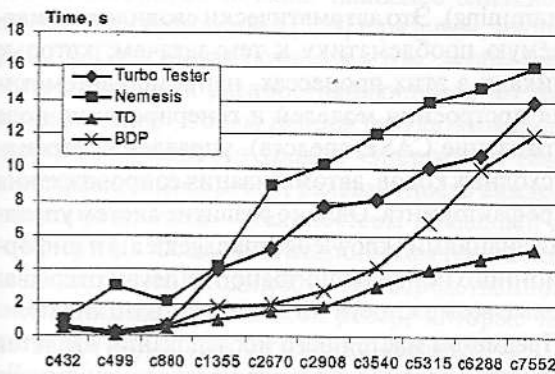


Рис. 8. Анализ быстродействия систем моделирования

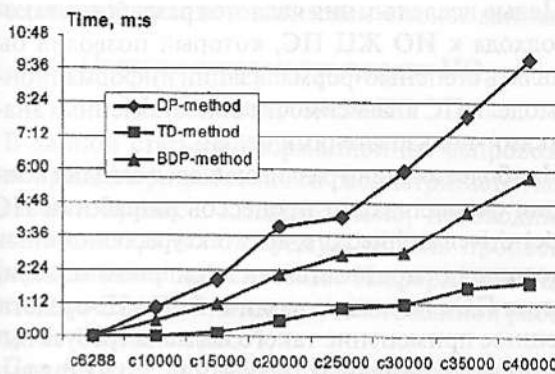


Рис. 9. Анализ быстродействия методов моделирования

Таким образом, основным результатом данной работы является усовершенствование дедуктивно-параллельного метода моделирования неисправностей цифровых систем, заключающееся в:

- 1) создании обобщенной модели процесса дедуктивно-параллельного анализа цифровых схем, имеющей вычислительную сложность  $n^2$  от числа линий схемы;
- 2) разработке дедуктивного алгоритма структурно-функционального анализа цифровых устройств в целях определения сходящихся разветвлений и декомпозиции структуры для раздельного моделирования СР и древовидных структур;
- 4) разработке топологического алгоритма моделирования неисправностей для древовидной структуры цифрового устройства, имеющего линейную вычислительную сложность от числа линий схемы.

Литература: 1. Wang X., Hill F.G., Mi Zh. A sequential circuit fault simulation by surrogate fault propagation // Proc. 1989 IEEE International test conference, IEEE Computer society, 1989. P. 9-18. 2. Nishida T., Miyamoto S., Kozawa T., Satoh K. RFSIM: Reduced fault simulator // IEEE Transactions on computer-aided design. 1987. Vol. CAD-6, No 3. P. 392-402. 3. Hahanov V.I., Babich A.V., Hyduke S.M. Test Generation and Fault Simulation Methods on the Basis of Cubic Algebra for Digital Devices. Proceedings of the Euromicro Symposium on Digital Systems Design DSD2001. Warsaw, Poland. September, 4-6, 2001. P. 228-235. 4. Хаханов В.И., Хак Х.М. Джахирул, Масуд М.Д. Мехеди. Модели анализа неисправностей цифровых систем на основе FPGA, CPLD // Технология и конструирование в электронной аппаратуре. 2001. № 2. С. 3-11. 5. Levendel Y.H., Menon P.R. Comparison of fault simulation methods – Treatment of unknown signal values // Journal of digital systems. 1980. Vol. 4. P. 443-459. 6. Abramovici M., Breuer M.A. and Friedman A.D. Digital systems testing and testable design. Computer Science Press. 1998. 652 p. 7. Убар П.П. Анализ диагностических тестов для комбинационных цифровых схем методом обратного прослеживания неисправностей // Автоматика и телемеханика. 1977. №8. С.168-176. 8. Автоматизированное проектирование цифровых устройств / С.С.Бадулин, Ю.М.Барнаулов и др./ Под ред. С.С. Бадулина. М.: Радио и связь. 1981. 240 с. 9. Active-HDL User's Guid. Second Edition. Copyright. Aldec Inc. 1999. 213p. 10. Горбатов В.А. Основы дискретной математики. М.: Высш. шк., 1986. 311 с. 11. Hahanov V.I., Sysenko I.Y., Skvortsova O.B. Test Generator Used Genetic Algorithms and Reconfigurable Deductive-Parallel Fault Simulation Method for Digital Devices. – 9-th International Conference. Mixed Design of Integrated Circuits and Systems. – Poland. – Wroclaw – 2002. – P. 549-554. 12. V. Hahanov, R. Ubar and S. Hyduke. Back-Traced Deductive-Parallel Fault Simulation for Digital Systems. – Proceedings of Euromicro Symposium on Digital Systems Design. – Belek-Antalya, Turkey. – 2003. – P. 370-377.

Поступила в редколлегию 11.11.2004