

КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ И ТЕХНИЧЕСКАЯ ДИАГНОСТИКА



УДК 519.713:681326

ВDP-МЕТОД МОДЕЛИРОВАНИЯ НЕИСПРАВНОСТЕЙ ДЛЯ СИНТЕЗА ТЕСТОВ ЦИФРОВЫХ ПРОЕКТОВ

*ХАХАНОВ В.И., КОЛЕСНИКОВ К.В.,
ХАХАНОВА А.В.*

Предлагается быстродействующий дедуктивно-параллельный метод обратного моделирования неисправностей, использующий процедуру суперпозиции решений, ориентированный на обработку сверхбольших цифровых проектов вентиляльного и регистрового уровней описания. Приводятся структуры данных и программно-ориентированные алгоритмы для реализации метода в составе автоматической системы генерации тестов.

1. Введение

Актуальность работы определяется необходимостью значительного улучшения средств моделирования и генерации тестов для структурно- и функционально-сложных цифровых систем, имплементированных в кристаллы программируемой логики. Существующие автоматические системы тестирования известных фирм: Cadence, Mentor Graphics, Synopsys, Logic Vision [www.cadence.com, www.logicvision.com, www.simucad.com, www.synctest.com, www.synopsys.com, www.mentorgraphics.com] ориентированы на обработку кристаллов размерностью около 100 тыс. вентилялей за приемлемое время, составляющее несколько часов. Но уже сейчас ясно, что данные средства синтеза тестов и моделирования неисправностей могут быть неприемлемыми относительно времени обработки чипов, насчитывающих несколько миллионов вентилялей. Нужны принципиально новые подходы, позволяющие на порядок повысить быстродействие анализа цифровой системы на стадии ее проектирования в целях построения тестов верификации. Один из них, решающий проблему создания быстродействующего метода моделирования одиночных константных дефектов для оценки качества тестов проверки неисправностей проектируемых цифровых систем на основе программируемой логики, предлагается в данной работе.

Объект тестирования – цифровая система, имплементируемая в кристаллы программируемой логики и представленная в форме булевых уравнений, которые записаны на языке VHDL.

Цель исследования – разработка метода моделирования одиночных константных неисправностей для оценки качества синтезируемых тестов цифровых систем, имплементируемых в ПЛИС, содержащих миллионы вентилялей. Практически приемлемым следует считать время моделирования, линейно-зависящее от числа эквивалентных линий, если количество сходящихся разветвлений составляет не более десяти процентов от общего числа.

Задачи исследования:

1. Создание обобщенной модели процесса дедуктивно-параллельного анализа цифровой схемы на основе процедуры обратной суперпозиции.
2. Разработка алгоритмов структурно-функционального анализа цифровых систем в целях определения множества сходящихся разветвлений и реконфигурации структуры для реализации процедуры суперпозиции.
3. Создание внутренней интерпретативно-компилятивной модели цифрового устройства для эффективного исправного анализа логических элементов и их неисправностей одиночного константного типа.

4. Алгоритмическая реализация метода моделирования неисправностей на основе реконфигурирования модели устройства в целях существенного уменьшения времени оценки качества тестов.

Основу ОДП-метода (Backtraced Deductive-Parallel) – Обратное Дедуктивно-Параллельное моделирование неисправностей – составляют: кубическое моделирование дефектов [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 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 [1]:

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

где $F = (F_1, F_2, \dots, F_i, \dots, F_n)$ – модель исправного поведения схемы, включающей n линий и/или функциональных элементов; $T = (T_1, T_2, \dots, T_t, \dots, T_k)$ – тест длиной k двоичных наборов; $L = (L_1, L_2, \dots, L_t, \dots, L_k)$ – множество дедуктивных функций (ДФ) параллельного моделирования неисправностей на тесте T , соответствующих модели исправного поведения F . Компонент функционального описания цифровой системы $F_i \in F$ представляет собой булеву функцию для вычисления состояния i -й линии:

$$F_i = f_i(X_{i1}, X_{i2}, \dots, X_{ij}, \dots, X_{in_i}). \quad (2)$$

В качестве результата исправного моделирования функции f_i выступают значения координаты $T_{ti} \in T_t$, причем $T_{ti} = F_i$ на тесте t . При этом имеется в виду, что тест есть матрица исправного поведения цифровой системы

$$T = [T_{ti}] = (T_{t1}, T_{t2}, \dots, T_{ti}, \dots, T_{tn}). \quad (3)$$

С учетом разбиения теста на составляющие векторы уравнение (1) получения ДФ для $T_t \in T$ принимает следующий вид: $L_t = T_t \oplus F$. Если функциональное описание цифровой системы представлено компонентами, формирующими состояния всех линий схемы, то в качестве формулы преобразования исправной модели примитива в дедуктивную функцию выступает выражение

$$L_{ti} = T_t \oplus F_i = f_i[(X_{i1} \oplus T_{t1}), (X_{i2} \oplus T_{t2}), \dots, (X_{ij} \oplus T_{tj}), \dots, (X_{in_i} \oplus T_{tn_i})] \oplus T_{ti}, \quad (4)$$

которое по существу аналогично формуле дедуктивного анализа цифровых схем, представленной в [5]. Практическая реализация выражения (4) оформляется в следующий алгоритм анализа дефектов.

1. Формирование интерпретативных моделей цифровой системы $W = \{F, L_0\}$. Определение начального значения тест-вектора $t=0$.

Инициализация векторов проверенных на тесте

$$T = [T_{ti}] \text{ дефектов } \bigvee_{i=1}^n (D_i^0 = 0; D_i^1 = 0).$$

2. Определение номера очередного входного набора $t=t+1$ для $T_t \in T$. Если входных наборов нет ($t > k$) – конец моделирования.

3. Исправное моделирование всех примитивов (невходных линий) $F_i (i = \overline{1, n})$ цифровой схемы на входном наборе $T_t^X \in T_t$ с использованием модели $F \in W$ в целях доопределения невыходных координат вектора $T_t^{\overline{X}} \in T_t$:

$$T_t^{\overline{X}} = f(T_t^X, F). \quad (5)$$

Идентичность векторов исправных состояний линий в двух соседних итерациях $T_t^r = T_t^{r-1}$ является условием перехода к следующему пункту.

Для моделирования последовательностных схем и организации событийности используется анализ пары соседних векторов (T_{t-1}, T_t) . Примитив $F_i (i = \overline{1, n})$ моделируется, если на линиях выполняется условие $[T_{t-1}^X(F_i) \neq T_t^X(F_i)]$ – наличие изменений на входах рассматриваемого элемента.

4. Инициализация матрицы проверяемых на тест-векторе дефектов $M = [M_{ij}]$ в соответствии с выражением

$$[M_{ij}]_{(i, j = \overline{1, n})} = \begin{cases} 0 & \Leftarrow (i \neq j); \\ 1 & \Leftarrow (i = j). \end{cases} \quad (6)$$

Инициализация векторов проверяемых на тест-векторе дефектов $\bigvee_{i=1}^n (S_i^0 = 0; S_i^1 = 0)$.

Реконfigurирование примитивов $L_i (i = \overline{1, n})$ интерпретативной модели $L_i \in W$ на основе применения формулы (4) для текущего вектора исправного состояния

$$T_t = (T_{t1}, T_{t2}, \dots, T_{ti}, \dots, T_{tn})$$

в целях получения модификации $L_{ti} = T_t \oplus F_i$.

5. Формирование невыходных строк матрицы проверяемых неисправностей путем их параллельного моделирования с помощью примитивов $L_{ti} \in L_t$.

6. Формирование объединенного вектора проверяемых неисправностей S путем применения формулы

$$S = \bigvee_{i \in Y} M_i^r \quad (7)$$

ко всем строкам матрицы, соответствующим выходным наблюдаемым линиям схемы.

7. При выполнении условия $\bigvee_{i=1}^n (S_i = S_i^0 \vee S_i^1)$ осуществляется вычисление качества тестового набора по формуле

$$Q(T_t) = \frac{1}{2n} \left[\sum_{i=1}^n (S_i^0 + S_i^1) \right] \quad (8)$$

и переход к следующему пункту, иначе – формирование пары $\{S^0, S^1\}$ по выражению

$$S^0 = S \wedge T, (S_i^0 = S_i \wedge T_i); S^1 = S \wedge \overline{T}, (S_i^1 = S_i \wedge \overline{T}_i) \quad (9)$$

или, если наблюдается исчезновение проверяемых неисправностей в векторе S по отношению к паре $\{S^0, S^1\}$: $\bigvee_{i=1}^n [(S_i = 0) \& (S_i^0 \vee S_i^1 = 1)]$, выполняется исключение таких дефектов из процесса моделирования по правилу

$$(S_i^0 = S_i^1 = 0) \Leftarrow \bigvee_{i=1}^n [(S_i = 0) \& (S_i^0 \vee S_i^1 = 1)].$$

Переход к п. 5.

8. Формирование векторов проверенных неисправностей в соответствии с выражением

$$D^0 = D^0 \vee S^0, D^1 = D^1 \vee S^1 \quad (10)$$

и вычисление качества теста по формуле

$$Q(T) = \frac{1}{2n} [\sum_{i=1}^n (D_i^0 + D_i^1)]. \quad (11)$$

Переход к п. 2.

Предложенная алгоритмическая реализация ориентирована как на табличное описание примитивов произвольной сложности RTL уровня, так и на вентиляльное представление цифровых систем. Быстродействие алгоритма практически инвариантно к компилятивным и интерпретативным моделям цифровых устройств, однако чисто интерпретативная реализация является более технологичной с позиции программирования.

3. Интерпретативная модель анализа исправного поведения примитива

Структурная модель логического анализа примитивного элемента цифрового устройства представлена на рис. 1.

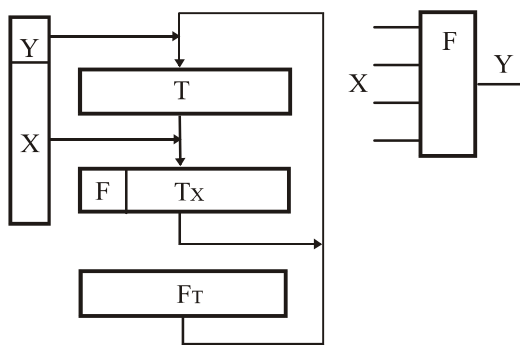


Рис. 1. Структурная модель анализа элемента

Здесь речь идет о процедуре определения состояния координаты тест-вектора $T(Y)$, соответствующей выходу Y логического элемента F , по его входным значениям, представленным вектором $X = (X_1, X_2, \dots, X_k)$, при условии, что F_T есть таблица истинности многообразия булевых функций, определенная на векторе двоичных переменных $(F^* T_X)$:

$$T(Y) = F_T(F^* T_X) = F_T(F^* T(X_1) * T(X_2) * \dots * T(X_k)).$$

Иначе, чтобы определить состояние координаты $T(Y)$, необходимо сформировать двоичный вектор состояния входных переменных T_X на основе вектора номеров линий X (см. рис.1). Затем следует выполнить конкатенацию полученного вектора с двоичным кодом типа примитива (функции) F в целях получения строки $(F^* T_X)$ обобщенной таблицы истинности F_F , где в столбце Y , соответствующем значению функции, находится искомое состояние координаты $T(Y)$. Модель дедуктивно-параллельного анализа неисправностей, кроме структуры, участвующей в исправном моделировании, содержит два дополнительных модуля (M, F_F), как показано на рис.2.

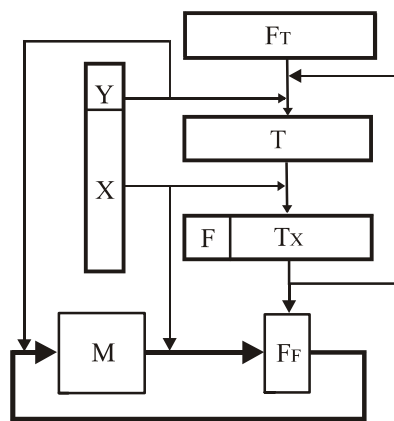


Рис. 2. Модель дедуктивно-параллельного анализа
Аналитическое выражение для вычисления векторов проверяемых неисправностей, объединенных в матрицу M , с помощью дедуктивной функции F_F , являющейся модификацией F_T по выражению (4), имеет вид:

$$M(Y) = F_F(F^* T(X), M(X_1) \circ M(X_2) \circ \dots \circ M(X_k)) = F_{(F^* T(X))}(M(X_1) \circ M(X_2) \circ \dots \circ M(X_k)).$$

Здесь операция, обозначенная символом $\circ = \{\wedge, \vee\}$, может быть представлена дизъюнкцией или конъюнкцией; $F_{(F^* T(X))}$ – дедуктивный элемент, определяемый двоичным словом-адресом $(F^* T(X))$.

Для определения состояния вектор-строки $M(Y)$ необходимо установить адрес (тип) дедуктивной компилятивно реализуемой функции, используя полученную для исправного моделирования конкатенацию двоичных последовательностей $(F^* T(X))$. Входные переменные для элемента $F_{(F^* T(X))}$ являются регистровыми, теоретическая размерность которых равна числу линий в цифровом устройстве. Далее осуществляется последовательное выполнение $(k-1)$ регистровых операций над входными векторами $M(X_i) \in M$. Результат в виде последовательности $M(Y)$ заносится в матрицу M . Векторная переменная X_i может иметь знак инверсии. Тогда перед выполнением операции $\circ = \{\wedge, \vee\}$ осуществляется инверсия содержимого регистровой переменной:

$$M(\bar{X}_i) = \bar{M}(X_i).$$

В качестве примера рассмотрим обобщенную таблицу истинности, чтобы определить состояние исправного поведения и выбрать двоичный адрес дедуктивного функционального элемента анализа неисправностей:

S	F	X ₁ X ₂	Y	F _F
∧	00	00	0	00
	00	01	0	01
	00	10	0	10
	00	11	1	11
∨	01	00	0	11
	01	01	1	10
	01	10	1	01
	01	11	1	00
∧	10	00	1	00
	10	01	1	01
	10	10	1	10
	10	11	0	11
∨	11	00	1	11
	11	01	0	10
	11	10	0	01
	11	11	0	00

Здесь столбец F – код функции исправного поведения, (X₁, X₂) – двоичные входные наборы таблицы истинности каждой из четырех функций, Y – столбец состояния исправного поведения выхода функций, F_F – код адреса компилятивной модели дедуктивного элемента, которая представлена четырьмя примитивами:

$$F_F = \begin{cases} 00 \rightarrow X_1 \wedge X_2; \\ 01 \rightarrow X_1 \wedge \overline{X_2}; \\ 10 \rightarrow \overline{X_1} \wedge X_2; \\ 11 \rightarrow X_1 \vee X_2. \end{cases}$$

Вычислительная сложность обработки цифровой схемы, состоящей из n двухвходовых вентилях, определяется выражением

$$Q = [(2K + A) + A + (2n\tau) / W] = \\ = [2(K + A) + (2n\tau) / W] \times n,$$

где K – время, затрачиваемое на конкатенацию битов для получения адреса состояния выхода примитива; A – время выборки содержимого ячейки (бита) по его адресу; τ – время выполнения регистровой операции (and, or, not); W – разрядность регистра.

Если учесть, что первое слагаемое 2(K+A) является несущественным по сравнению со вторым, то вычислительная сложность будет представлена формулой

$$Q = (2n^2\tau) / W.$$

Таким образом, затраты времени обработки цифрового устройства пропорциональны квадрату числа вентилях.

4. ОДП-метод моделирования неисправностей

Предложенная интерпретативно-компилятивная модель дедуктивно-параллельного анализа неисправностей и исправного поведения является базовой для ОДП-метода и гарантирует нахождение решения в виде множества всех дефектов, проверяемых на тест-векторе за n² итераций. Для понижения вычислительной сложности нахождения решения предлагается стратегия моделирования, представленная на рис.3.



Рис. 3. Стратегия ОДП-метода моделирования

Основная идея повышения быстродействия моделирования неисправностей связана с преобразованием сходящихся разветвлений в псевдовыходы в целях последующего применения процедуры суперпозиции для древовидных структур и их необработки в случае фиксации непроверяемости линий сходящихся разветвлений.

Стратегия ОДП-метода моделирования неисправностей цифрового устройства с предварительным структурным анализом (см. рис. 3) включает следующие шаги:

1. Идентификация линий сходящихся разветвлений, инвариантных по отношению к тест-векторам. Вычислительная сложность данной процедуры

$Q_1 = n^2$, но она выполняется на стадии предварительного анализа и практически не влияет на быстродействие моделирования тест-векторов.

2. Моделирование неисправностей линий сходящихся разветвлений на тест-векторе. Модификация схемной структуры путем преобразования сходящихся разветвлений в псевдовыходы цифрового устройства.

3-4. Вычисление линий подграфов схемы, моделирование неисправностей которых на тест-векторе не должно проводиться вследствие существования формального доказательства их непроверяемости.

5. Определение фрагментов графа цифрового устройства, корректных для выполнения суперпозиции решений на тест-векторе.

6-8. Выполнение процедуры суперпозиции векторов проверяемых неисправностей примитивов на скорректированной модели цифрового устройства.

Интерес представляет алгоритм определения сходящихся разветвлений на основе анализа графовой структуры цифрового устройства. Сложность выполнения алгоритма равна квадрату числа линий в схеме. Для комбинационных схем основные шаги представлены пунктами:

1. Пусть все линии цифрового устройства можно разбить на подмножества:

$$V = (V^Y, V^S, V^R),$$

где Y – идентификатор линии, относящейся к внешним выходам; S – обозначение линии, имеющей одного преемника, соединенной с одним элементом; R – идентификатор линии разветвления, имеющей более одного элемента-преемника.

Определение множества преемников для каждой линии схемы путем формирования вектора числа преемников:

$$V = (V_1, V_2, \dots, V_i, \dots, V_n),$$

$$V_i = \begin{cases} 0 \leftarrow V_i \in V^Y; \\ 1 \leftarrow V_i \in V^S; \\ \geq 2 \leftarrow V_i \in V^R \end{cases}$$

2. Выбор очередной линии $V_i \in V^R$ для определения ее принадлежности к множеству сходящихся разветвлений $V_i^R \in V^{RC}$. Данная процедура выполняется путем логического моделирования графовой структуры от линии $V_i^R \in V$ на множестве всех ее преемников до внешних выходов схемы. Первоначально все линии обнуляются $\forall_{i=1}^n V_i = 0$.

3. Присвоение рассматриваемой линии разветвления значения 1: $V_i^R \in 1$. После этого реализуется последовательность операций $V_i = V_i + 1$ для всех линий, являющихся преемниками V_i^R . Если на некотором шаге будет зафиксирован результат $V_i = V_i + 1 = 2$, являющийся критерием сходимости для разветвления V_i^R , то оно заносится в список $V_i^R \in V^{RC}$ и осуществляется переход к п. 2.

Повторение п. 2 и 3 выполняется для всех линий разветвления.

В качестве примера структурного анализа выступает схема, представленная на рис. 4, где линии 15,

17, 19 есть сходящиеся разветвления. В результате применения алгоритма их поиска схема реконфигурируется в четыре древовидных структуры, определяемые подграфами с корневыми вершинами, являющимися выходами или псевдвыходами устройства (рис.5). Моделирование неисправностей такой схемы с помощью обратной суперпозиции требует уже линейных затрат памяти и времени в функции от числа эквипотенциальных линий и квадратичных затрат для обработки сходящихся разветвлений:

$$Q = (r^2 / W) + n_r + n_p + (n - r - r^0),$$

где (r^2 / W) – время моделирования неисправностей r сходящихся разветвлений, число которых определяется как $r = 0.2 \times n$; $n_r = n$ – время реконфигурирования примитивов схемы на входном наборе; $n_p = n$ – время поиска подграфов линий, соответствующих непроверяемым сходящимся разветвлениям; $(n - r - r^0) = n - 0.2 \times n - 0.4 \times n = 0.4 \times n$ – время выполнения процедуры суперпозиции на множестве линий схемы без сходящихся разветвлений и предшественников для непроверяемых сходящихся разветвлений. Учитывая фактические значения указанных параметров в функции от числа линий схемы, можно получить следующую оценку быстродействия ВДР-метода:

$$Q = [(0.2 \times n)^2 / W] + n + n + (n - 0.2 \times n - 0.4 \times n) =$$

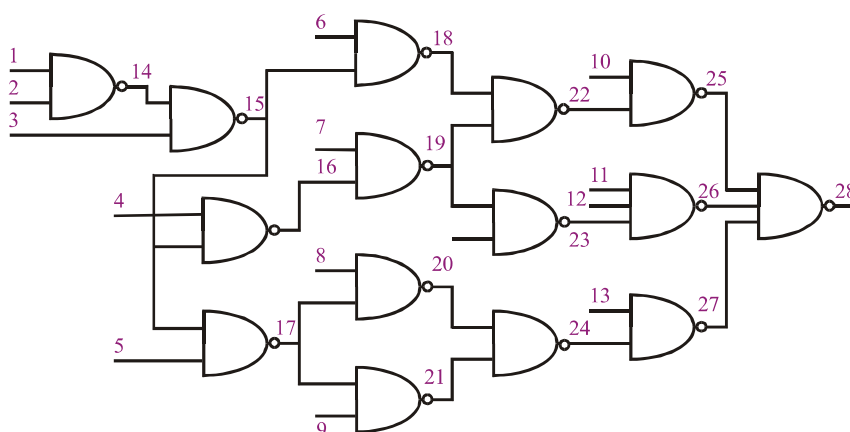
$$= [(0.2 \times n)^2 / W] + 2.4 \times n.$$

Таким образом, выигрыш в быстродействии предложенного метода тем больше, чем меньше процент сходящихся разветвлений в схеме цифрового устройства.

С учетом предварительного вычисления сходящихся разветвлений алгоритмическая реализация ОДР-метода представлена следующими пунктами:

1. Фаза прямого исправного моделирования цифровой схемы. Предназначена для определения реакции всех невходных линий схемы на тест-вектор

$$T_t \in T = [T_{ti}].$$



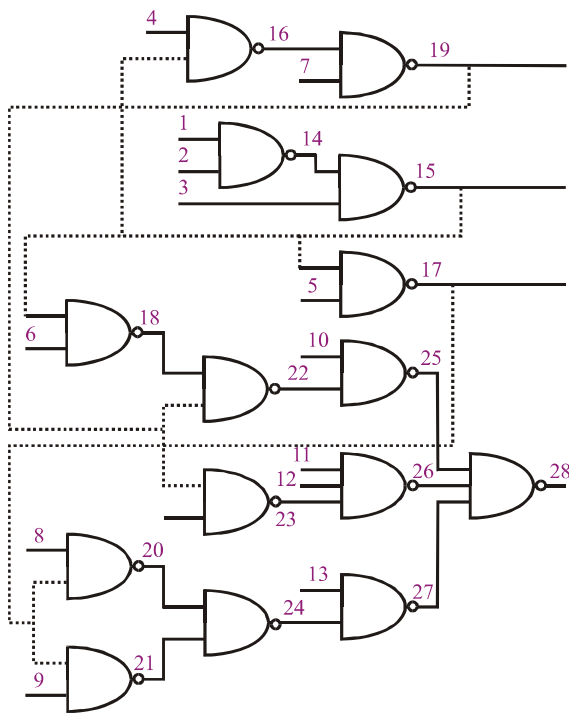


Рис. 5. Древоподобные фрагменты схемы

Множество всех линий дифференцируется на входные, внутренние и выходные: (X, Y, Z) . Это означает, что тест-строка (-столбец) из матрицы исправного поведения T представляется как $T_t = (T_t^X, T_t^Z, T_t^Y)$. То же самое относится и к вектору моделирования неисправностей $S = (S^X, S^Z, S^Y)$, который каждый раз строится для новой строки T_t .

2. Инициализация вектора моделирования неисправностей цифровой схемы $S = (S_i^X = 0, S_i^Z = 0, S_i^Y = 1)$. Единичное значение координаты вектора $S_i = 1$ является индикатором проверки одиночной неисправности, инверсной двоичному исправному состоянию линии T_{ij} .

3. Дизъюнкция вектора проверяемых входных дефектов i -го примитива $S^i = (S_1^i, S_2^i, \dots, S_j^i, \dots, S_{n_i}^i)$ с вектором S моделирования неисправностей схемы при условии, что на линии, соответствующей выходу i -го элемента, имеется единичное значение $S_i = 1$:

$$S(I_j^i) = S(I_j^i) \vee_{j=1}^{n_i} S_j^i \Leftarrow S_i = 1,$$

где $I^i = (I_1^i, I_2^i, \dots, I_j^i, \dots, I_{n_i}^i)$ – вектор номеров входных линий i -го примитива. Последний анализируется на основе применения дедуктивно-параллельного алгоритма к матрице проверяемых дефектов, но не схемы, а рассматриваемого элемента. Такой анализ можно выполнять и дедуктивно, используя собственные входные списки проверяемых дефектов примитива.

1. Определение двоичного вектора линий сходящихся разветвлений

$$R = (R_1, R_2, \dots, R_i, \dots, R_n),$$

$$R_i = \begin{cases} 1 \leftarrow R_i \in R^{RC}; \\ 0 \leftarrow R_i \notin R^{RC}, \end{cases}$$

относительно которого выполняется процедура дедуктивно-параллельного анализа.

2. Генерирование исходных списков неисправностей линий сходящихся разветвлений схемы

$$\{S_j\} = \begin{cases} j: \bar{T}_{ij} \leftarrow R_j = 1; \\ \emptyset \leftarrow R_j = 0, \end{cases}$$

где $\{S_j\} \subseteq S; j = \bar{1}, n; S$ – вектор списков дефектов.

3. Моделирование неисправностей линий сходящихся разветвлений $L_R \subseteq L = (L_R, L_{\bar{R}})$ дедуктивно-параллельным или дедуктивным методом на реконфигурируемой модели устройства, соответствующей тест-вектору T_t . Использование упомянутого метода обусловлено незначительным ($\approx 20\%$) числом сходящихся разветвлений, поскольку $L_R / L_{\bar{R}} \ll 1$.

4. Исключение из процесса моделирования неисправностей древоподобных подграфов с корневой вершиной, являющейся сходящимся разветвлением, неисправность которого не проверяется:

$$L_U = L_U \setminus L_R^0 \cup f^{-1}(L_R^0).$$

5. Моделирование неисправностей линий, дополняющих сходящиеся разветвления и подграфы с непроверяемыми корневыми вершинами до полного множества $L_U = L_U \setminus [L_R^1 \cup L_R^0 \cup f^{-1}(L_R^0)]$ на основе процедуры суперпозиции. Анализ проверки дефектов выполняется только относительно выходных наблюдаемых линий схемы, дополненных разветвлениями, неисправности которых обнаруживаются на тест-векторе T_t :

$$L_Y = L_Y \cup L_R^1 \subseteq L_R = \{L_R^0, L_R^1\}; |L_R| = r,$$

где L_Y, L_R^0, L_R^1 – выходные линии схемы, разветвления с непроверяемыми и обнаруживаемыми неисправностями соответственно.

Последовательно рассматриваются все линии, которые являются предшественниками для полученного множества $L_Y = L_Y \cup L_R^1$. Относительно каждой линии из списка L_Y выполняется процедура суперпозиции для входного тестового набора.

5. Заключение

Предложенный метод моделирования неисправностей ориентирован на обработку сверхсложных цифровых систем на основе ПЛИС, содержащих миллионы вентилях. Тестовые эксперименты программной реализации метода на нескольких десятках цифровых комбинационных и последователь-

ностных схем дали хорошие результаты по быстродействию в сравнении с традиционными алгоритмами параллельного и дедуктивного анализа. Отдельные примеры сравнения быстродействия разработанного метода и существующих базовых представлены на рис. 6. Ускорение моделирования составляет не менее десяти раз. Таким образом, основным результатом данной работы является усовершенствование дедуктивно-параллельного метода [1-3] моделирования неисправностей цифровых систем, заключающееся в: 1) создании обобщенной модели процесса дедуктивно-параллельного анализа цифровой схемы на основе процедуры обратной суперпозиции, имеющей линейную вычислительную сложность от числа линий схемы; 2) разработке алгоритмов структурно-функционального анализа цифровых систем в целях определения множества сходящихся разветвлений и реконфигурации структуры для реализации процедуры суперпозиции; 3) создании внутренней интерпретативно-компилятивной модели цифрового устройства для эффективного исправного анализа логических элементов и их неисправностей одиночного константного типа.

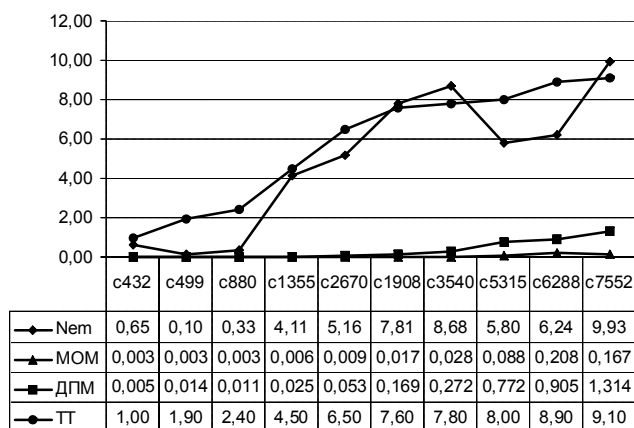


Рис. 6. Анализ быстродействия: Nem – система Nemesis; ТТ– Turbo Tester; ДПМ – дедуктивно-параллельный метод; MOM – BDP-метод обратного моделирования

Литература: 1. 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. 2. 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. 3. Хаханов В.И., Сысенко И.Ю., Хак Х.М. Джахирул, Масуд М.Д. Мехеди. Кубическое моделирование неисправностей цифровых проектов на основе FPGA, CPLD // Радиоэлектроника, информатика, управление. 2001. № 1. С. 123-129. 4. Levedel 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. 5. Abramovici M., Breuer M.A. and Friedman A.D. Digital System Testing and Testable Design, Computer Science Press, 1998. 652 p. 6. Хаханов В.И. Техническая диагностика элементов и узлов персональных компьютеров. К.: ИЗМН. 1997. 308 с. 7. Убар Р.Р. Анализ диагностических тестов для комбинационных цифровых схем методом обратного прослеживания неисправностей // Автоматика и телемеханика. 1977. №8. С.168-176.

Поступила в редколлегию 16.05.2002

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Хаханов Владимир Иванович, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика вычислительных устройств, систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26. E-mail: hahanov@kture.kharkov.ua

Колесников Константин Васильевич, ст. преподаватель кафедры КС Черкасского государственного технологического университета. Научные интересы: тестирование цифровых систем и сетей. Увлечения: живопись – постимпрессионизм, поэзия – серебряный век, символизм. Адрес: Украина, 18006, Черкассы, бульв. Шевченко, 460, тел. 43-74-28, e-mail: kvvk601@mail.ru

Хаханова Анна Владимировна, студентка ХНУРЭ. Научные интересы: моделирование и диагностика цифровых устройств. Увлечения: аэробика, литература. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26.

УДК 681.325

ПОЭЛЕМЕНТНЫЙ КОНТРОЛЬ СУММАТОРА С ПЛАВАЮЩЕЙ ТОЧКОЙ

ДРОЗД А.В., БАДР Я.

Предлагается метод поэлементного функционального контроля сумматора с плавающей точкой, использующий естественные ресурсы организации и функционирования одноктактных вычислительных устройств для выполнения приближенных вычислений. Метод обеспечивает построение экономичной самопроверяемой схемы контроля с возможностью обнаружения ошибок в зависимости от их величины и частоты появления.

1. Введение

Одним из важнейших требований, предъявляемых к современным вычислительным устройствам (ВУ), является достоверность их функционирования, которая обеспечивается средствами и методами функционального диагностирования, выполняющими проверку работоспособности ВУ в процессе их использования по назначению [1].

Данное требование необходимо обеспечить в совокупности с другими требованиями, такими как производительность и диапазон представления чисел. В настоящее время наиболее распространены одноктактные ВУ, в которых производительность достигается с использованием матричного распараллеливания вычислений. Расширение диапазона представления чисел привело к переходу от естественной формы к форматам с плавающей точкой.