

## Эволюционные методы генерации проверяющих тестов для неконстантных неисправностей цифровых схем

Ю.А. Скобцов<sup>1</sup>

<sup>1</sup>Санкт-Петербургский политехнический университет

**1. Введение.** Диагностика современных цифровых систем и их элементной базы требует анализа не только константных неисправностей, но и более адекватных моделей физических дефектов. Вследствие технологического прогресса, который ведет к увеличению плотности на кристалле, росту числа пересечений проводящих слоев и повышению рабочей частоты, тестирования классических константных неисправностей в настоящее время явно недостаточно, поскольку необходимо анализировать физические дефекты, которые влияют на временные характеристики схемы [1]. Особенно это характерно для глубокого субмикронного (deep submicron - DSM) проектирования.

Увеличение числа транзисторов на кристалле ведет к тому, что большее количество элементов переключаются одновременно, что может уменьшить для них уровень напряжения и увеличить задержки распространения сигналов. При этом некоторые пересекающиеся линии, которые предполагались электрически изолированными, могут взаимодействовать друг с другом. Одно из подобных взаимодействий, вызванное паразитической емкостной связью между проводниками, называется “crosstalk” (перекрестная помеха), может привести к функциональным проблемам и ухудшить временные характеристики.

Проблема построения тестов для индуцированных (“crosstalk”) неисправностей является одной из самых сложных в диагностике цифровых схем и привлекает внимание многих исследователей [2]. Обычно рассматриваются два основных типа перекрестных помех: 1) “crosstalk” индуцированные импульсы; 2) “crosstalk” индуцированные задержки. В первом случае быстро переключающаяся линия - «агрессор» (aggressor) может индуцировать короткий импульс на статической линии - «жертве» (victim). Второй случай (индуцированные задержки) имеет место тогда, когда на линиях «агрессоре» и «жертве» происходят (почти) одновременные переходы сигналов. Сильный «агрессор» может вызвать задержку распространения сигнала на линии-«жертве», которая имеет противоположное значение сигнала. Если на линиях «агрессоре» и «жертве» происходят переходы сигналов в противоположных направлениях, то время перехода увеличивается и имеет место эффект «перекрестного замедления» (“crosstalk slowdown”).

**2. Перекрестные неисправности.** Рассмотрим цифровую схему, которая имеет перекрестную неисправность на некоторой линии. Для генерации теста для перекрестной неисправности необходимо:

- Найти значения входов, которые дают необходимые значения сигналов на линии - агрессоре;
- Найти значения входов, гарантирующие необходимые значения статического сигнала на линии-жертве;
- Найти значения входов, обеспечивающие распространение cross эффекта от линии-жертвы к внешнему входу.

Рассмотрим задачу построения теста для неисправности «индуцированная» задержка, которая вызывается влиянием нескольких линий-агрессоров. В этом случае линии-жертвы входят в некоторый путь, связывающий внешний вход с внешним выходом схемы. Множество линий-агрессоров образуют те линии схемы, которые могут воздействовать на линии-жертвы и тем самым вызвать задержку распространения сигналов на указанном пути.

При решении этой задачи генерация тестов для неисправностей «индуцированная задержка» со многими агрессорами необходимо решить как минимум три подзадачи:

- 1) выбор множества критических путей, формирующий линии-жертвы;
- 2) выбор множества линий-агрессоров для заданного критического пути;

- 3) построение пары входных тестовых наборов, проверяющих индуцированные задержки для заданного пути и множества линий-агрессоров.

Выбор множества линий-агрессоров для данной линии-жертвы можно формализовать следующим образом. Пусть  $A=\{a_1, a_2, \dots, a_m\}$  обозначает множество возможных линий агрессоров. Для каждой линии-агрессора  $a_i$  определим  $c_i$  - степень его влияния на жертву, которая определяется прежде всего величиной емкостной связи между этими линиями.

Далее определим множество булевых переменных  $S=\{s_1, s_2, \dots, s_m\}$ , где каждая переменная  $s_i$  соответствует своему агрессору  $a_i$ . При этом

$$s_i = \begin{cases} 1, & \text{если } a_i \text{ принадлежит текущему множеству агрессоров} \\ 0, & \text{иначе} \end{cases} .$$

То есть каждая переменная  $s_i$  является признаком принадлежности линии  $a_i$  к рассматриваемому в текущий момент множеству агрессоров. Тогда степень влияния текущего

множества агрессоров на линию-жертву можно определить следующим образом  $\sum_{i=1}^m s_i \cdot c_i$  .

Для максимального увеличения кросс-индуцированной задержки текущее множество агрессоров должно быть возбуждено таким образом, чтобы эффект задержки на линии-жертве был максимальным. То есть необходимо найти такой булев вектор  $S$ , при котором

$$\text{имеет место } \max \sum_{i=1}^m s_i \cdot c_i$$

и выполняются необходимые логические условия для построения тестовой пары. Очевидно, что данная задача носит переборный характер и является NP-трудной. В худшем случае необходимо рассмотреть все  $2^m$  подмножеств линий агрессоров.

**3. Моделирование.** Моделирование перекрестных неисправностей выполняется на основе интерпретативного моделирования в многозначном алфавите  $S_8$  [1]. Цель моделирования – проверка обнаружимости перекрестной неисправности данной входной последовательностью. Укрупненный алгоритм моделирования неисправностей представлен ниже.

Моделирование перекрестных неисправностей (комбинационная логическая схема, входная последовательность input sequence T, пара линий (агрессор, жертва)) {

```

Ввод описания схемы;
Инициализация схемы;
Ввод входной последовательности T;
Ввод перекрестной неисправности;
FOR каждого набора входной последовательности T {
Моделирование набора в алфавите  $S_8$ ;
If (выполняется условие возникновения перекрестной неисправности) {
Ввод эффекта перекрестной неисправности и активация для данной пары
(агрессор, жертва);
Ввод входного набора и моделирование;
If (эффект неисправности наблюдается на выходе) данная перекрестная неисправность
обнаруживается;
}
}
}

```

Ввод положительных и отрицательных индуцированных импульсов должен быть выполнен в процессе моделирования неисправностей.

Пример 1 многозначного моделирования в алфавите  $S_8$  представлен на рис.1 для схемы C17 из каталога ISCAS85. Положительный индуцированный импульс рассматривается между линией-агрессором 19GAT(7) и линией-жертвой 16GAT(8). Отметим, что здесь представлены значения для одной пары наборов и данной неисправности (Рис.1).

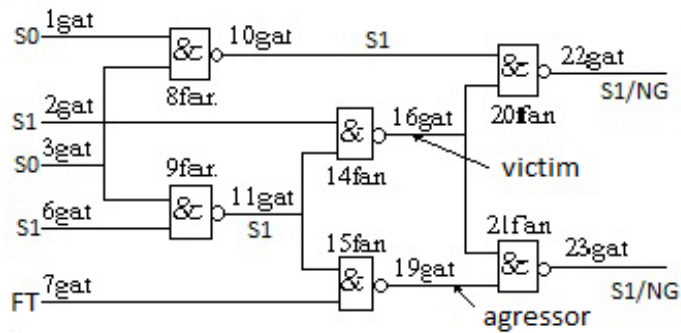


Рис.1. Логическое моделирование положительного импульса

Пример 2 индуцированной задержки для схемы С17 между линией-жертвой h линией-агрессором i представлен на рис.2. В этом случае устанавливается передний фронт RT на линии-агрессоре и задний фронт FT на линии-жертве.

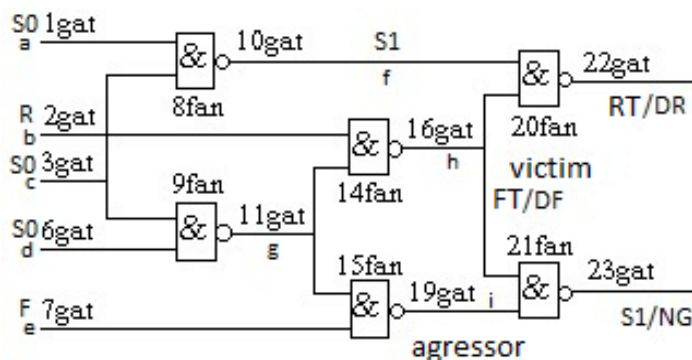


Рис.2. Логическое моделирование индуцированной задержки

**4. Генетический алгоритм генерации теста.** Генетический алгоритм и многозначная логика с успехом могут быть использованы в процессе генерации тестов [2]. Генетический алгоритм обеспечивает механизм случайного направленного поиска тестовых пар, которые удовлетворяют заданным условиям.

Предложен эволюционный алгоритм (ЭА) для генерации проверяющих тестов для неисправностей типа “crosstalk”. На наш взгляд применение эволюционных методов при построении проверяющих тестов для таких неисправностей еще более оправдано, чем для классических константных неисправностей[1]. Можно провести аналогию с решением задач численной оптимизации, где ЭА целесообразно использовать, прежде всего, там, где не работают классические градиентные методы. Следует отметить, что ЭА часто позволяют задачу синтеза, в некотором то смысле, свести к задаче анализа. Если есть средства анализа (моделирования поведения исправной или неисправной цифровой системы), то ЭА обеспечивают целенаправленный случайный поиск решения задачи синтеза. При этом задачи синтеза и анализа могут выполняться на различных уровнях. Это позволяет строить тесты для новых типов неисправностей (практически для произвольных неисправностей, для которых есть соответствующая модель).

Классический ГА использует три основных оператора: репродукция, кроссинговер и мутация. Благодаря этим операторам популяция (множество особей – потенциальных решений данной проблемы) эволюционирует от одного поколения к другому.

В разработанном генетическом алгоритме генерации тестов начальная популяция  $N$  порождается случайно. Здесь каждая особь  $(x_1, x_2, \dots, x_n)$  имеет длину  $2n$  битов, где  $n$  – число внешних входов схемы. При этом каждому внешнему входу  $x_i \in B_2 \times B_2$  соответствует два связанных двоичных бита, которые для данного входа представляют значения сигналов на двух последовательных наборах (00), (11), (10) и (11), что в многозначном алфавите моделируется значениями S0, S1, R и F соответственно. Множество пар входных наборов составляет популяцию.

Генерация новой популяции производится с помощью следующих генетических операторов. Используется турнирный метод отбора родителей. При турнирном отборе [1,3] выбираются случайно  $m$  особей, затем лучшая из них отбирается в качестве потенциального партнера. Процедура продолжается до тех пор, пока не сформируется промежуточная популяция. Здесь параметр отбора  $2 \leq m \leq N$  ( $m=2,3$ ). Затем выполняется кроссинговер с высокой вероятностью  $P_c$ . В качестве кроссинговера мы применяем однородный кроссинговер[1,3].

Построенные потомки мутируют с малой вероятностью  $P_m \ll 1$  и внедряются в текущую популяцию. При этом используется классический оператор мутации. Но он применяется к к группе из 2-х бит соответственно значениям S0, S1, RT and FT.

При этом отбор родителей смещается в сторону наборов с лучшей полнотой покрытия неисправностей. Фитнесс-функция является линейной комбинацией трех компонент:

$$f_f = w_e f_e + w_p f_p + w_c f_a,$$

где  $f_e$  оценивает способность набора возбуждать необходимые значения на линии-жертве,  $f_p$  оценивает способность особи распространять кросс эффект до внешних выходов,  $f_a$  оценивает его способность учитывать влияние линии-агрессора. При оценке значений фитнес-функции используются результаты многозначного моделирования в алфавите  $S_8$ .

Следующее поколение генерируется на основе текущей популяции с использованием тех же генетических операторов. Процесс повторяется до тех пор, пока не выполнен критерий останова.

**5. Эксперименты.** Результаты генерации тестов для одиночных перекрестных неисправностей типа положительные и отрицательные индуцированные импульсы и задержки представлены в табл.1.

Таблица 1. Генерация тестов для одиночных перекрестных неисправностей

Схема	Fault coverage			
	Положительный импульс PG	Отрицательный импульс NG	Задержка DR	Задержка DF
C432	98.05	99	100	98.01
C499	99	84.81	99.4	99.8
C880	97.09	95.13	98.16	98.85
C1355	88.15	94.41	99.18	99.43
C1908	89.75	96.77	95.63	93.88
C2670	90.31	94.15	94.71	90.45
C5315	99.70	99.87	99.71	99.72
C6288	99.91	98.97	99.31	99.32
C7552	97.23	98.28	99.05	97.03

Кроме одиночных перекрестных неисправностей эксперименты проведены также для неисправностей со многими агрессорами. Результаты представлены в табл.2.

Видно, что в среднем генетический алгоритм обеспечивает 2-кратное увеличение полноты по сравнению со случайным методом. В тоже время для комбинационных схем полнота покрытия неисправностей составляет 70%. Заметим, что для некоторых схем (например, S526, S420.1, S1238) псевдо-случайный метод не может построить тесты. Для последовательностных схем результаты хуже, чем средняя полнота составляет около 40%. Но следует отметить, многие из неисправностей, отмеченных как целевые, (ряд которых показан во 2-м столбце табл.6) реально не существуют, вследствие противоречивых логических условий. Поэтому реальная полнота для перекрестных неисправностей может быть выше.

**Заключение.** Рассмотрена проблема построения тестов для индуцированных неисправностей типа импульс и задержка. Показано, что дальнейшее развитие многозначного моделирования позволяет моделировать и этот класс неисправностей. Модифицирован генетический алгоритм для генерации пар тестовой последовательности для проверки неисправностей индуцированная задержка. Рассмотрена также проблема построения тестов для индуцированных задержек со многими агрессорами.

Таблица 2. Генерация тестов для перекрестных неисправностей со многими агрессорами

Схема	Число целевых неисправностей	Полнота покрытия %		Время (sec)
		Псевдослучайный	Генетический алгоритм	
C17	42	45,23	62.05	0.18
C432	9327	59.52	70.15	241.14
C449	21879	30.27	71.81	591.17
C880	9279	2.31	47.53	628.14
S27	74	6.78	37.87.	0.39
S208	743	30.96	32.43	6.42
S208.1	558	1.63	45.21	12.72
S298	537	43.04	62.58	9.43
S344	1190	64.38	66.14	18.53
S349	1197	30.51	57.92	19.24
S526	891	0.3	19.23	21.34
S386	4195	15.81	25.51	58.04
S510	1098	36.83	43.87	14.73
S420.1	1276	0.3	36.93	63.95
S820	7738	26.34	35.41	166.42
S1196	10630	0.3	14.61	782.45
S1238	5822	0.2	13.51	531.23
S1488	4305	18.43	22.45	184.52
S1494	4283	18.23	18.43	165.42
average		22.70	41.24	185.02

Разработан программный модуль, где реализованы многозначное моделирование и генетический алгоритм для построения тестов для не классических (константных) неисправностей, который интегрирован в систему моделирования и генерации тестов АСМИДА. Выполнены тестирование и верификация эффективности разработанных программных модулей для схем международных каталогов ISCAS85 и ISCAS89, которые подтвердили достигнутые высокие характеристики.

#### Литература

1. Скобцов Ю.А., Сперанский Д.В., Скобцов В.Ю. Моделирование, тестирование и диагностика цифровых устройств.-М.:Национальный открытый университет "ИНТУИТ", 2012. – 439 с.
2. Skobtsov Yu. A, Skobtsov V. Yu. Evolutionary test generation methods for digital devices // Design of Digital Systems and Devices / [eds.: M.Adamski et al.]. – Berlin: Springer-Verlag, 2011, – P.331-361. – (Lecture Notes in Electrical Engineering, Vol. 79).
3. Скобцов Ю. А., Сперанский Д.В. Эволюционные вычисления: учебное пособие.-М.: Национальный Открытый Университет "ИНТУИТ", 2015. – 331с.