

Эволюционные алгоритмы построения проверяющих тестов для перекрестных неисправностей цифровых схем

Ю.А. Скобцов

Санкт-Петербургский государственный университет аэрокосмического приборостроения,

г. Санкт-Петербург, ya_skobtsov@list.ru

Аннотация — Рассмотрены основные модели перекрестных неисправностей (crosstalk faults): индуцированные положительные и отрицательные импульсы, индуцированные задержки. Получил развитие метод моделирования перекрестных неисправностей в 9-значном алфавите. На этой основе разработан генетический алгоритм построения проверяющих тестов для одиночных перекрестных неисправностей. Рассмотрена задача генерации проверяющего теста для неисправности типа «индуцированная задержка». Предложен генетический алгоритм построения проверяющих тестов для неисправностей индуцированная задержка со многими линиями-агрессорами.

Ключевые слова — генерация тестов, перекрестные неисправности, генетический алгоритм, многозначная логика, моделирование неисправностей.

I. ВВЕДЕНИЕ

Тестирование современных компьютерных систем и их элементной базы требует анализа не только классических константных неисправностей, но и более адекватных моделей поведения цифровых схем на высоких частотах. При возрастании плотности на кристалле, количества пересечений проводящих слоев и тактовой частоты, тестирования классических модельных неисправностей (константных) недостаточно, поскольку необходимо проверять временные характеристики схемы [1]. Это особенно актуально для глубокого субмикронного (deep submicron - DSM) проектирования.

Повышение плотности транзисторов на кристалле может привести к тому, что большее число логических элементов переключаются почти одновременно. При этом для них возможны падение уровня напряжения и замедление распространения сигналов. В этом случае некоторые пересекающиеся линии, предполагавшиеся электрически изолированными, могут влиять друг на друга. Подобное взаимодействие, которое вызвано паразитной емкостной связью между проводниками, называют перекрестной помехой - "crosstalk". Это явление часто ведет к функциональным проблемам и ухудшает временные характеристики цифровых схем.

Задача генерации проверяющих тестов для индуцированных неисправностей ("crosstalk") - одна из самых тяжелых в тестировании цифровых схем и ей

посвящены работы многих исследователей [2]. Как правило, рассматриваются две перекрестные помехи: 1) "crosstalk" индуцированные импульсы; 2) "crosstalk" индуцированные задержки. При наличии первой помехи переключающаяся линия - «агрессор» (aggressor) индуцирует короткий импульс на статической линии - «жертве» (victim). Вторая помеха возникает при (почти) одновременном изменении сигналов на линиях «агрессоре» и «жертве». В этом случае сильный «агрессор» может вызвать уменьшение времени распространения сигнала на линии-«жертве», имеющей противоположное значение сигнала. Тогда наблюдается эффект «перекрестного замедления» ("crosstalk slowdown").

II. АНАЛИЗ ПРОБЛЕМЫ

Работы по моделированию и тестированию перекрестных неисправностей проводятся с 90-х годов и актуальны для теории и практики надежного проектирования и тестирования современных цифровых устройств. Они основаны на применении различных подходов, таких как временной анализ, многозначные логики, метод критических путей, модификации метода PODEM, детерминированные структурные и основанные на моделировании методы [1-3]. Так в [2] предложен метод, который основан на поиске пары входных наборов, устанавливающих необходимое значение сигнала на линии-«жертве» и вызывающее изменение сигнала на линии-«агрессоре», который обеспечивает проявление неисправности на одном из внешних выходов. В этой работе рассматриваются индуцированные импульсы "crosstalk".

Авторы [3-5] разработали смешанный генератор тестовых сигналов XGEN для индуцированных задержек "crosstalk". Здесь предложен смешанный метод построения тестов, где в процессе поиска двоичных наборов применяется аналоговое моделирование, что позволяет точнее оценить задержки распространения сигналов. Статический временной анализ дает возможность рассчитать временные окна для входных и выходных сигналов элементов. Искомое временное окно определяется в результате пересечения временных окон линий «агрессора» и «жертвы». В этом случае для заданной неисправности - пары «агрессор-жертва» алгоритм

дает возможность определить необходимые значения сигналов на взаимодействующих линиях схемы и распространить влияние неисправности хотя бы до одного внешнего выхода. При этом в процессе построения тестов применяется 11-значный алфавит и модификация метода PODEM на этапе доопределения (justification).

В [6] модель задержки пути используется в комбинации с методом критических путей. При этом в качестве основы используется метод генерации тестов для неисправностей типа «задержка распространения сигнала» без фазы доопределения. Далее на этапе доопределения применяется эволюционный алгоритм, где в процессе генерации теста используются временные характеристики. Этот подход получил развитие в [7] для решения задачи генерации тестов в случае многих «агрессоров», воздействующих на определенный путь. При этом строится граф импликации, где учитывается логическая и структурная информация схемы для проверки конфликтных ситуаций в процессе построения тестов. При генерации тестовых наборов применяется модификация метода PODEM.

В работе [8] рассматривается тестирование индуцированных задержек. Метод генерации тестов основан на построении критических путей на основе статического временного анализа схемы. При этом при проверке чувствительности путей используется критерий устойчивой тестируемости. Для найденного чувствительного пути активируются пары «агрессор-жертва» так, чтобы максимально усилить воздействие агрессора на путь и увеличить задержку распространения сигнала вдоль этого пути.

В [9] используется генератор тестов с использованием единичной модели задержки. При этом активизируется путь, обеспечивающий необходимые условия переходов сигналов. Здесь известная модель неисправности типа «задержка» распространения сигналов используется для тестирования индуцированных задержек.

III. ПЕРЕКРЕСТНЫЕ НЕИСПРАВНОСТИ

Пусть цифровая схема имеет перекрестную неисправность на некоторой паре линий. При генерации теста для перекрестной неисправности необходимо:

- найти значения входных сигналов, обеспечивающие требуемые значения сигналов на линии-агрессоре;
- найти значения входных сигналов, гарантирующие требуемые значения статического сигнала на линии-жертве;
- найти значения входных сигналов, которые обеспечивают распространение кросс-эффекта от линии-жертвы к внешнему входу.

Рассмотрим задачу построения теста для неисправности «индуцированная задержка» со многими линиями-агрессорами. При этом 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 увеличить задержку распространения сигналов на указанном пути.

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

- определение множества критических путей, которое формирует линии-жертвы;
- определение множества линий-агрессоров для заданного критического пути;
- построение пары входных тестовых наборов, которая проверяет индуцированные задержки для заданного пути и построенного множества линий-агрессоров.

Отбор 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 данную 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 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.

IV. ЛОГИЧЕСКОЕ МНОГОЗНАЧНОЕ МОДЕЛИРОВАНИЕ

Для перекрестных неисправностей моделирование выполняется на основе логического моделирования с применением многозначного алфавита C_9 [1]. Целью моделирования является проверка обнаружимости перекрестной неисправности порожденной входной последовательностью. Укрупненный алгоритм моделирования неисправностей представлен ниже.

Моделирование перекрестных неисправностей (схема, входная последовательность, 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  $C_8$ ;
    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 выполняться в процессе логического моделирования неисправностей.

Пример моделирования в многозначном алфавите C_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).

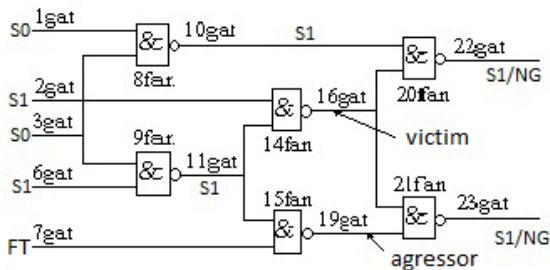


Рис.1. Логическое моделирование положительного импульса

Пример 2 индуцированной задержки для схемы C17 между линией-жертвой h и линией-агрессором i представлен на рис. 2. При этом на линии-агрессоре устанавливается передний фронт RT и на линии-жертве задний фронт FT.

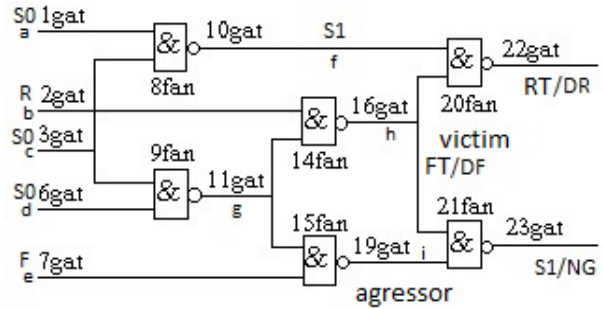


Рис.2. Логическое моделирование индуцированной задержки

V. ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ГЕНЕРАЦИИ ТЕСТОВ

Генетический алгоритм (ГА) в сочетании с многозначной логикой с успехом используется для построения тестов цифровых схем [2]. В данном случае ГА дает механизм случайного направленного поиска тестовых пар, удовлетворяющих заданным условиям.

Разработан эволюционный алгоритм (ЭА) для построения проверяющих тестов для перекрестных неисправностей. Применение эволюционных методов для генерации проверяющих тестов для перекрестных неисправностей более оправдано, чем для классических модельных неисправностей типа константа [1]. Здесь можно провести аналогию с задачами численной оптимизации, где ЭА имеет смысл использовать прежде всего там, где неэффективны классические градиентные методы. Отметим, что ЭА часто позволяют задачу синтеза свести к задаче анализа. При наличии средств анализа (логического моделирования поведения исправной или неисправной цифровой системы) ЭА обеспечивают целенаправленный случайный поиск решения задачи синтеза. Отметим, что в этом случае задачи синтеза и анализа могут выполняться даже на различных уровнях (например, на логическом и схемном). Это дает возможность строить тесты для новых типов неисправностей (практически произвольная модель), что важно для новых технологий.

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

В предложенном ГА начальная популяция N порождается случайно. При этом каждая особь (потенциальный тест для данной неисправности) (x_1, x_2, \dots, x_n) имеет размер $2n$ битов, где n – число внешних входов схемы. Здесь каждому внешнему входу схемы $x_i \in B_2 \times B_2$ соответствует два связанных двоичных бита, которые для этого 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 подобных пар наборов составляет популяцию.

Генерация новой популяции следующего поколения выполняется с помощью следующих генетических операторов. Применяется турнирный метод отбора родительских особей [1, 3]. В этом случае выбираются случайно m особей, далее лучшая из них отбирается в качестве потенциальной родительской особи. Эта процедура (турнир) повторяется до тех пор, пока не сформируется промежуточная популяция. Используются следующие значения параметров отбора $2 \leq m \leq N$ ($m=2, 3$). Далее выполняется однородный генетический оператор кроссинговера с высокой вероятностью P_c [1, 3].

Затем порожденные потомки мутируют с малой вероятностью $P_m \ll 1$ и внедряются в текущую популяцию. Здесь применяется модификация классического оператора мутации. Этот оператор применяется к группе из 2-х бит соответственно к значениям S0, S1, RT and FT.

В этом случае отбор родителей смещается в сторону тестовых наборов с лучшей полнотой покрытия неисправностей. Фитнесс-функция оценивает качество потенциальных решений – тестов. В данном случае она является линейной комбинацией трех компонент:

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

где компонента f_e оценивает способность набора устанавливать требуемые значения на линии-жертве, компонента f_p оценивает способность особи распространять кросс эффект до внешних выходов и компонента f_a оценивает его способность учитывать влияние линии-агрессора. При оценке значений фитнес-функции используются результаты многозначного моделирования в алфавите S_9 . Следующее поколение генерируется на основе текущей популяции с использованием тех же генетических операторов. Процесс повторяется до тех пор, пока не выполнен критерий останова.

VI. ИНДУЦИРОВАННЫЕ ЗАДЕРЖКИ СО МНОГИМИ АГРЕССОРАМИ

Определение множества критических путей. Количество потенциальных перекрестных неисправностей – пар линий жертва-агрессор для реальной схемы огромно. Отметим, что большую часть таких неисправностей не имеет смысла или невозможно тестировать. Выбор критических путей в схеме представляет собой серьезную задачу, которой посвящено достаточно много публикаций [1].

Поэтому сначала определяется сокращенное множество неисправностей индуцированных задержек, как правило, на основе статического временного анализа схемы [5]. В этом случае необходимо выполнить следующие действия.

1. Для каждой линии схемы необходимо найти «временное окно», определяемое самым ранним и самым поздним возможным временем изменения сигнала.
2. Далее из максимальных значений самых поздних времен изменения сигналов следует найти самый долгий (критический) путь в схеме. Тогда линии, входящие в этот путь образуют потенциальные «жертвы» для перекрестных неисправностей.
3. Затем временное окно для каждой линии – жертвы необходимо сравнить с временным окном потенциальной линии - агрессора. В случае пересечения эта пара жертва-агрессор заносится во множество перекрестных неисправностей.

Определение множества целевых неисправностей. Далее для линии-жертвы, вошедшей в критический путь, находятся линии-агрессоры схемы, которые могут индуцировать задержку распространения сигнала на критическом пути. Даже для одного критического пути число возможных линий-агрессоров для реальной схемы может быть очень велико и зависит экспоненциально от числа линий в данной схеме. Поэтому необходимо понизить сложность задачи генерации тестов для кросс-индуцированных задержек. Для этого необходимо отбросить линии, которые не могут влиять на линии жертвы, входящие в данный критический путь, и оставить множество целевых неисправностей, с учетом которых и будет строиться проверяющий тест для этого пути.

Сократить множество неисправностей можно на основании следующей информации[5]:

- 1) пространственные соотношения линий в схеме;
- 2) временные соотношения;
- 3) функциональные соотношения.

Генетический алгоритм генерации теста. Ясно, что проверяющий тест для перекрестных неисправностей типа индуцированные задержки должен состоять из пар входных наборов, обеспечивающих приведенные выше условия. Заметим, что для данного критического пути время распространения сигнала может быть различным для разных пар входных наборов. Очевидно, что желательно найти пары наборов с максимальной временной задержкой, которые позволяют проверить целевые неисправности. В качестве особи популяции, как и ранее, целесообразно использовать пару входных наборов $(x_1^t, x_2^t, \dots, x_n^t; x_1^{t+1}, x_2^{t+1}, \dots, x_n^{t+1})$ с общей длиной $2n$ битов, где каждому i -ому внешнему входу схемы соответствует два связанных двоичных бита $(x_i^t; x_i^{t+1})$, которые для данного входа представляют значения сигналов на двух последовательных наборах (00), (11), (10) и (11). Множество таких пар входных наборов составляет

популяцию, которая развивается с помощью генетических операторов репродукции, кроссинговера и мутации.

При отборе родительских особей также используется турнирный метод [6]. Как и ранее, применяются два способа выбора: детерминированный и случайный. При детерминированном способе отбор родителя выполняется с вероятностью, равной 1. При использовании случайного метода выбор родительской особи выполняется с вероятностью $P_i < 1$. Как правило, популяция разбивается на туры по 2-3 особи в каждой ($m=2, 3$).

Далее, как обычно, к родительским особям применяется генетический оператор кроссинговера. Здесь также используется модификация однородного кроссинговера [6], которая учитывает структуру хромосомы, где каждому входу схемы в паре входных наборов соответствует 2 связанных бита ($x_i^t; x_i^{t+1}$). При этом значения 2 битов потомков, соответствующие определенному входу, формируются из аналогичных битов этого же входа одного из родителей. В этом случае случайно генерируется двоичная маска кроссинговера той же длины (с тем же числом бит), что у хромосом родителей, где значение бита маски (0 или 1) показывает родителя, из которого копируется ген потомка.

Кроме однородного кроссинговера можно использовать также и структурный кроссинговер, где обмен между родителями производится столбцами, соответствующими одной древовидной подсхеме. При этом на этапе препроцессорной обработки схема должна быть разбита на древовидные подсхемы. Тогда внешние входы схемы, которые «питают» одну древовидную подсхему, относятся к одной и той же группе. В этом случае обмен производится группами столбцов, соответствующих одной и той же древовидной подсхеме. Отметим, что в этом случае в одну группу попадают входы, определяющие значения внутренних «узловых» точек схемы. Особенно это касается линии-жертвы, значения сигналов которой определяется только значениями внешних входов соответствующей древовидной подсхемы. Тогда обмен генетическим материалом выполняется с большей направленностью для соответствующих внутренних линий схемы, что в некоторых случаях повышает эффективность поиска проверяющих тестовых последовательностей.

После выполнения кроссинговера к полученным потомкам применяется генетический оператор мутации, где также учитывается структура хромосомы пары входных наборов, используемая в данной задаче. При этом случайно выбирается вход схемы (не 1 бит хромосомы!) и с небольшой вероятностью инвертируются значения двух битов, которые соответствуют выбранному входу схемы.

При этом значение фитнес-функции определяется количеством проверяемых перекрестных

неисправностей, которые можно получить с помощью программы моделирования неисправностей.

VII. 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			
	Полож. импульс PG	Отрицат. 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

Таблица 2

Генерация тестов для неисправностей задержка со многими линиями-агрессорами

Схема	Число целевых неисправностей	Полнота покрытия %		Время (sec)
		ПС	ГА	
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

VIII. ЗАКЛЮЧЕНИЕ

Разработаны многозначные модели для логических элементов, что позволяет эффективно моделировать

перекрестные неисправности. Предложен модифицированный генетический алгоритм для генерации пар тестовых наборов, проверяющих одиночные индуцированные импульсы и одиночные неисправности типа индуцированная задержка. Показано, что дальнейшее развитие многозначного моделирования позволяет моделировать и этот класс неисправностей. Модифицирован генетический алгоритм генерации пар тестовой последовательности для проверки неисправностей индуцированная задержка со многими агрессорами.

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 АСМИД-А.

Выполнены тестирование и верификация эффективности разработанных программных модулей для схем международных каталогов ISCAS85 и ISCAS89, которые подтвердили достигнутые высокие характеристики.

ЛИТЕРАТУРА

- [1] Ю.А. Скобцов, Д.В. Сперанский, В.Ю. Скобцов Моделирование, тестирование и диагностика цифровых устройств.–М.:Национальный открытый университет “ИНТУИТ”, 2012. – 439 с.
- [2] A. Rubio, N.Itazaki, X.Xu, K.Kinoshita /An approach to the analysis and detection of crosstalk faults in digital VLSI circuits// IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 1994. – Vol.13, №3. – P.387-394.
- [3] W. Y. Chen, S. K. Gupta, M. A. Breuer. Analytic Models for Crosstalk Delay and Pulse Analysis under Non-Ideal Inputs//International Test Conference. – Washington, DC, Nov.1997:proceedings.-Los Alamitos: :IEEE,1997. – P. 809-818.
- [4] W.Y.Chen, S.K.Gupta,M.A.Breuer. Test generation for Cross-Induced Delay in Integrated Circuits //International Test Conference.- Atlantic City, NJ, USA, 28-30 September 1999. : Proceedings.-Los Alamitos: :IEEE, 1997.-P.191-200.
- [5] W. Y. Chen, S. K. Gupta, M. A. Breuer. Test generation for Cross-Induced Faults: Framework and computational results // Journal of Electronic Testing: Theory and Applications. – 2002. – Vol.16. – P.17-28.
- [6] A. Krstic, J.-J.Liou, Y.-M.Jiang, K.-T. Cheng. Delay Testing Considering Cross-Induced Effects// International Test Conference. –Baltimore, MD, 30 Oct-01 Nov 2001: proceedings.-Los Alamitos: :IEEE, 2001.– P.558-567.
- [7] X.Bai, S.Dey, A.Krstic. A Hybrid Structural SAT Based ATPG for Crosstalk / // International Test Conference. – Washington, DC,-Sept. 30-Oct. 2, 2003:proceedings.-Los Alamitos: :IEEE. – 2003.– P.112-121.
- [8] A. Arunachalam, R.Arunachalam. A Novel Algorithm for Testing Crosstalk Induced Delay Faults in VLSI Cicuits // International Conference on VLSI Design.-Kolkata, India .- 3-7 January 2005:proceedings.-Los Alamitos: IEEE. – 2005. - P.479-484.
- [9] H. Li, X. Li Selection of Crosstalk-induced Faults in Enhanced Delay test // Journal of Electronic Testing: Theory and Applications. – 2005. – Vol. 21, № 2. – P. 181-195.
- [10] Ю. А. Скобцов, Д. В. Сперанский. Эволюционные вычисления – М.: Национальный Открытый Университет “ИНТУИТ”, 2015. – 331с.
- [11] Yu. A. Skobtsov, V. Yu. Skobtsov. 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).

Evolutionary Algorithms of Test Generation for Crosstalk Faults of Digital Circuits

Yu.A. Skobtsov

Saint Petersburg State University of Aerospace Instrumentation

Saint Petersburg, ya_skobtsov@list.ru

Abstract — Technological progress leads to density increasing on-chip, rising number of intersections in conductive layers and speeding-up of the clock frequency. As a result – the model of classical stuck-at faults is not enough effective and adequate because the analysis of physical defects affected on time parameters of a circuit is the need. Especially it is typical for deep submicron (DSM) designing. In the paper, there was considered the approach to pulse-induced crosstalk faults test generation. Besides it was considered the approach to multiple aggressor crosstalk faults test generation.

It is considered two basic types of crosstalk faults: crosstalk induced pulses; crosstalk induced delays. Under test generation for crosstalk faults it is necessary: find out input patterns, which cause required signal transient on aggressor line; find out input patterns guaranteeing necessary static signal on victim line and propagating arisen pulse from victim line to primary outputs. The solution of both problems is based on multi-valued logic and genetic algorithm application.

At that, the genetic algorithm provides a mechanism of the random direct search of test patterns pairs, which satisfy

noted conditions. To our point of view, application of evolutionary methods in test generation for considered faults is more justified than for classical stuck-at faults. It is necessary to note that genetic algorithms allow reducing synthesis problem to analysis problem (in some sense). Under the condition of analysis tools presence (fault-free or faulty circuit simulators) genetic algorithms provide direct random search for synthesis problem solution. To evaluate fitness-function value results of logical simulation in the multivalued alphabet is used [1]. In this way, we define newly detected faults for given input pattern pair (potential solution).

It was shown for combinational digital circuits that application of genetic algorithms and multivalued event-driven simulation in C8 alphabet allows effectively solve test generation problem for crosstalk faults and gives high fault coverage.

Keywords — test generation, crosstalk faults, genetic algorithm, multivalued modeling, fault simulation.

REFERENCES

- [1] Yu.A. Skobtsov, D.V. Speransky, V.Yu. Skobtsov. Modelirovanie, testirovanie i diagnostika tsifrovyyh ustroystv (Simulation, testing and diagnosis of digital devices).—M.:Natsionalniy otkrytyi universitet "INTUIT", 2012. — 439 c.
- [2] Rubio, N.Itazaki, X.Xu, K.Kinoshita //An approach to the analysis and detection of crosstalk faults in digital VLSI circuits// IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. — 1994. — Vol.13, №3. — P.387-394.
- [3] W. Y. Chen, S. K. Gupta, M. A. Breuer. Analytic Models for Crosstalk Delay and Pulse Analysis under Non-Ideal Inputs//International Test Conference. — Washington, DC, Nov.1997: proceedings.-Los Alamitos: :IEEE, 1997. — P. 809-818.
- [4] W.Y.Chen, S.K.Gupta, M.A.Breuer. Test generation for Cross-Induced Delay in Integrated Ciccuits //International Test Conference.- Atlantic City, NJ, USA, 28-30 September 1999. : proceedings.-Los Alamitos: IEEE, 1997.-P.191-200.
- [5] W. Y. Chen, S. K. Gupta, M. A. Breuer. Test generation for Cross-Induced Faults: Framework and computational results // Journal of Electronic Testing: Theory and Applications. — 2002. — Vol.16. — P.17-28.
- [6] Krstic, J.-J.Liou, Y.-M.Jiang, K.-T. Cheng. Delay Testing Considering Cross-Induced Effects// International Test Conference. —Baltimore, MD, 30 Oct-01 Nov 2001: proceedings.-Los Alamitos: IEEE, 2001.— P.558-567.
- [7] X.Bai, S.Dey, A.Krstic. A Hybrid Structural SAT Based ATPG for Crosstalk // International Test Conference. — Washington, DC,-Sept. 30-Oct. 2, 2003: proceedings.-Los Alamitos:IEEE. — 2003.—P.112-121.
- [8] Arunachalam, R.Arunachalam. A Novel Algorithm for Testing Crosstalk Induced Delay Faults in VLSI Ciccuits // International Conference on VLSI Design.-Kolkata, India .- 3-7 January 2005:proceedings.-Los Alamitos: IEEE. — 2005. - P.479-484.
- [9] H. Li, X. Li. Selection of Crosstalk-induced Faults in Enhanced Delay test//Journal of Electronic Testing: Theory and Applications.—2005.—Vol. 21, № 2. — P. 181-195.
- [10] Yu.A. Skobtsov, D.V. Speransky. Evolutsionnye vychisleniya (Evolutionary computations).—M.: Natsionalniy otkrytyi universitet "INTUIT", 2015. — 331c.
- [11] Yu. A. Skobtsov, V. Yu. Skobtsov. 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. — (Lecture2 Notes in Electrical Engineering, Vol. 79).