

УДК 681.32

## ГЕНЕРАЦІЯ ТЕСТОВ ДЛЯ ПЕРЕКРЕСТНИХ НЕИСПРАВНОСТЕЙ ТИПА ЗАДЕРЖКА

Ю.А. Скобцов, В.Ю. Скобцов, Нассер Іяд К.М.  
Донецкий национальный технический университет  
[skobtsov@kita.dgtu.donetsk.ua](mailto:skobtsov@kita.dgtu.donetsk.ua)

*Розглядається проблема побудови перевіряльник тестів для "перехресних несправностей" типу затримка, характерних для глибокого субмікронного проектування елементної бази сучасних комп’ютерних систем. При розв’язку поставленого завдання використовується багатозначне логічне моделювання й генетичний алгоритм генерації тестів, що перевіряють, для цих несправностей.*

Тестирование современных компьютерных систем требует анализа не только классических константных неисправностей, но и более сложных моделей, которые учитывают временные характеристики схемы. Особенно это характерно для глибокого субмікронного проектирования. К ним относятся «перекрестные неисправности» (cross talk faults)[1,2], где обычно рассматриваются два основных типа перекрестных помех: 1) “crosstalk” индуцированные импульсы; 2) “crosstalk” индуцированные задержки[3,4].

### **1. Неисправности индуцированные задержки**

В настоящей работе рассматриваются перекрестные неисправности, типа задержка при наличии которых имеет место эффект перекрестного замедления 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”). Если вызванный шум на линии-«жертве» больше порогового напряжения или индуцированная задержка больше допустимой, то это может привести к логическим отказам или функциональным проблемам на соседних триггерах или выходах. Перекрестные неисправности вызываются паразитическими наводками между соседними проводящими линиями, где доминируют емкостные составляющие.

## **2. Генерация тестов для неисправностей индуцированные задержки**

Генерация проверяющих тестов для этих неисправностей отличается от построения тестов для неисправностей типа задержка распространения сигналов несмотря на то, что она также основана на использовании пар тестовых наборов [5,6]. При построении проверяющего теста для такой неисправности необходимо: 1) найти входные наборы, которые вызывают необходимый переход сигналов на линии-агрессоре: 2) найти входные наборы, обеспечивающие необходимое изменение сигнала на линии-«жертве» и распространение возникшей задержки от жертвы до одного из внешних входов. Задача построения тестовой пары входных наборов для заданной пары линий схемы “жертва-агрессор” рассмотрена в предыдущих работах [3,4].

Здесь рассматривается задача построения теста для кратных неисправностей задержек, индуцированных различными линиями агрессорами. В этой постановке задачи 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 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.

## **3. Выбор множества критических путей**

Количество возможных перекрестных неисправностей – пар линий жертва-агрессор для реальной схемы очень велико. Но большинство таких неисправностей не имеет смысла или невозможно тестировать. Поэтому на первом этапе обычно находится сокращенное множество неисправностей индуцированных задержек на основе статического временного анализа схемы.

При этом необходимо выполнить следующие действия.

1. Для каждой линии схемы необходимо найти «временное окно», которое определяется самым ранним и самым поздним возможным временем изменения сигнала.
2. Из максимальных значений поздних времен изменения сигналов найти самый долгий (критический) путь в схеме. Линии,

входящие в этот путь образуют потенциальные «жертвы» для перекрестных неисправностей.

3. Временное окно для каждой линии – жертвы необходимо сравнить с с временным окном потенциальной линии-агрессора. Если эти окна пересекаются, то эта пара жертва-агрессор заносится в множество перкрестных неисправностей.

Например, для схемы С17 рис.1 на основе статического временного анализа можно найти 6 критических путей: [3gat, 11gat, 16gat, 22gat], [6gat, 11gat, 16gat, 22gat],[3gat, 11gat, 16gat, 23gat], [3gat, 11gat, 19gat,23gat,], [6gat, 11gat, 16gat, 23gat], [6gat, 11gat, 19gat, 23gat].

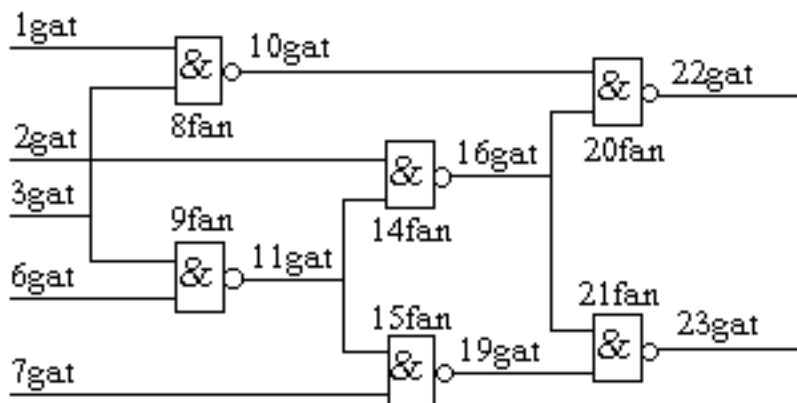


Рисунок 1 – Схема С17 каталога ISCAS85.

Таким образом, множество линий-жертв образуют следующие линии схемы рис.2 [3gat, 6gat, 11gat, 16gat, 19gat, 22gat, 23gat].

Всего для схемы рис.2 возможно 42 потенциальных перекрестных неисправностей. Рассмотрим пару входных наборов (00001, 01000). Для линии-жертвы 16gat можно выделить множество линий-агрессоров (10gat, 11gat, 19gat, 22gat, 23gat). Можно показать, что для приведенной пары входных наборов линии-агрессоры 19gat, 22gat проявляются на линии-жертве 16gat (перекрестные неисправности индуцированные задержки проверяются).

Таким образом, для каждой пары жертва-агрессор необходимо построить тестовую пару входных наборов, проверяющую данную неисправность.

#### 4. Моделирование перекрестных неисправностей типа задержка

Моделирование перекрестных неисправностей выполняется на основе событийного логического моделирования в многозначном алфавите [].

Его целью является проверка обнаружимости перекрестных неисправностей на заданной входной последовательности. Укрупненный алгоритм моделирования неисправностей состоит из следующих шагов.

1. Моделирование исправной схемы на случайной входной последовательности.
2. Выбор неисправности из множества перекрестных неисправностей.
3. Моделирование исправной и неисправной схемы для каждого входного набора.
4. Если линии жертва и агрессор имеют различные значения, то неисправность активируется.
5. Неисправность вносится в схему путем задержки сигнала на линии жертве на одну единицу модельного времени. Остальные события в неисправном варианте схемы такие же, как в исправной схеме.
6. Сравнение полученных в процессе моделирования значений сигналов исправной и неисправной схемы на внешних выходах схемы.
7. Если есть выход с различными значениями сигналов, то неисправность проверяется данной входной последовательностью и удаляется из множества неисправностей.
8. Переход на п.2.
9. Пока не достигнут конец, чтение следующего входного набора и переход на п.2.

### **5. Генетический алгоритм генерации теста**

Генетические алгоритмы широко применяются при построении проверяющих тестов для константных неисправностей более сложных моделей, например, задержек распространения сигналов[] в иску простоты, устойчивости и эффективности. На наш взгляд их применение при построении тестов для более сложных моделей неисправностей еще более оправдано, поскольку в этом случае трудно использовать детерминированные методы. Ясно, что проверяющий тест для перекрестных неисправностей типа индуцированные задержки должен состоять из пар входных наборов, обеспечивающих приведенные выше условия. Очевидно, что желательно найти пары наборов с максимальной задержкой, которые позволяют проверить целевые неисправности. Очевидно, что в качестве особи ГА здесь целесообразно использовать пару входных наборов, множество которых составляет популяцию. В качестве фитнес-функции проще всего использовать число проверяемых неисправностей, которое

можно получить с помощью программы моделирования неисправностей.

Псевдокод укрупненного ГА генерации проверяющих тестов представлен ниже, где вначале строится сокращенное множество перекрестных неисправностей типа задержка, затем в первой фазе генерируется начальная популяция тестовых наборов, которая используется во второй фазе.

```
{
  FR={сокращенное множество перекрестных неисправностей типа
  «задержка» };
  Начальная популяция=фаза_1(FR);
  if(FR=NULL)
    break;
  фаза_2(начальная популяция,FR)
}
```

При этом в первой фазе начальные входные последовательности генерируются псевдослучайным образом и далее для них выполняется моделирование перекрестных неисправностей. Если входная последовательность проверяет текущую неисправность, то эта неисправность исключается из множества обрабатываемых неисправностей а последовательность включается в множество решений – популяцию входных последовательностей Псевдокод укрупненного алгоритма первой фазы представлен ниже.

```
Фаза_1
Начальная популяция(FR);
For(i=0;i<max_it,i++)
{
  Начальная популяция=фаза_1(FR);
  Псевдослучайная генерация входных последовательностей
  длины l;
  for(каждой последовательности)
  {
    if( последовательность обнаруживает неисправность из FR)
    {
      Включение последовательности в популяцию;
      Исключение проверенной неисправности из FR;
    }
  }
}
Return(начальная популяция);
}
```

Во второй фазе, в качестве начальной популяции генетического алгоритма используется множество входных тестовых последовательностей, полученное в первой фазе. При выборе родительских особей применяется ранговый метод отбора. Для генерации новых особей используются стандартные операторы кроссинговера и мутации [7], среди которых мы отдаем предпочтение однородному кроссинговеру. Основные параметры ГА следующие: вероятность кроссинговера  $P_c=0.9$ , вероятность мутации  $P_m=0.01$ , мощность популяции  $N=16$ . Ниже приведен псевдокод укрупненного генетического алгоритма второй фазы генерации тестов.

```
Фаза_2
{
  Начальная популяция из первой фазы;
  for(p=0;p<max_gen,p++)
  {
    for(k=0;k<N,k++)
    {
      Выбор 2-х родительских особей;
      Выполнение кроссинговера с вероятностью  $P_c=0.9$ ;
      Выполнение мутации с вероятностью  $P_m=0.01$ ;
      Моделирование неисправностей;
      Вычисление фитнес-функции;
    }
    for(каждой последовательности)
    {
      if( последовательность обнаруживает неисправность из
FR)
      {
        Включение последовательности в популяцию;
        Исключение проверенной неисправности из FR;
      }
    }
  }
}
```

**Выводы.** Показано, что применение ГА в сочетании с логическим моделированием и временным анализом позволяет эффективно решать задачу построения тестов для неисправностей неисправностей индуцированных задержек.

#### Список литературы

1. Rubio. An approach to the analysis and detection of crosstalk faults in digital VLSI circuits/ Rubio, N.Itazaki, X.Xu, K.Kinoshita.-IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol.13,No.3,pp.387-394, March 1994

2. W.Y.Chen,S.K.Gupta,M.A.Breuer.Analytic Models for Crosstalk Delay and Pulse Analysis under Non-Ideal Inputs//Proc.of the Int’1.Test Conf., pp. 809-818, Nov.1997.
3. Ю.А.Скобцов,В. Ю. Скобцов, Нассер Іяд К.М. Генерация тестов для неисправностей типа индуцированные импульсы// Інформаційно-керуючі системи на залізничному транспорті.-№4.-2010.-С.27-29.
4. Ю.А.Скобцов,В. Ю. Скобцов, Нассер Іяд К.М. Проверяющие тесты cross talk неисправностей на основе эволюционных методов// Вестник национального технического университета «ХПИ», 2010.-№31.-С.170-176.
5. A.Krstic. Delay Testing Considering Cross-Induced Effects/ Krstic, J.-J.Liou, Y.-M.Jiang, K.-T.Cheng.-Proceedings of International Conference,2001.
6. Sunghoon Chun. XPDF-ATPG: An Efficient Test Pattern Generation for Crosstalk-Induced Faults/ Sunghoon Chun, Yongjoon Kim, Myuang-Hoon Yang, Sungho Kang.- 17<sup>th</sup> Asian Test Symposium, pp.83-88, 2008.
7. Ю.А.Скобцов, В.Ю.Скобцов. Логическое моделирование и тестирование цифровых устройств.-Донецк:ИПММ НАНУ, ДонНТУ, 2005.-436с.

Получено 05.09.2011