УДК 681.32

Ю.А. Скобцов, В.Ю. Скобцов, Ияд К.М. Нассер
Донецкий национальный технический университет
skobtsov@kita.dgtu.donetsk.ua

Построение тестов для перекрестных неисправностей типа задержка

Рассматривается проблема построения проверяющих тестов для «перекрестных неисправностей» типа задержка, характерных для глубокого субмикронного проектирования элементной базы современных компьютерных систем. При решении этой задачи используется многозначное логическое моделирование и генетический алгоритм генерации проверяющих тестов для этих неисправностей.

Перекрестные неисправности, проверяющие тесты, генетический алгоритм

Введение

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 глубокого субмикронного 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]

Неисправности индуцированные задержки

В настоящей работе рассматриваются перекрестные неисправности, типа задержка при наличии которых имеет место эффект перекрестного замедления исгналлов. 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.

Анализ литературы

Работы по моделированию и тестированию неисправностей типа "crosstalk" ведутся с середины 90-х годов. Так в [2] предложен метод, который основан на поиске пары входных наборов, которые устанавливают определенное значение сигнала на линии-«жертве» и вызывают переход на линии-«агрессоре», который обеспечивает распространение влияния неисправности до одного из внешних выходов. В этой работе рассматриваются "crosstalk" индуцированные импульсы.

Авторы [3-5] разработали смешанный генератор тестовых сигналов XGEN для "crosstalk" индуцированных задержек. Они предложили смешанный метод генерации тестов, где при поиске двоичных наборов используется аналоговое моделирование, которое позволяет оценить задержки распространения сигналов. Используя преобразование Лапласа они получили выражения для неисправностей "crosstalk" в s-области, которые используются для анализа зависимости атрибутов импульса в схеме со сосредоточенными параметрами и временных характеристик фронтов. Статический временной анализ позволяет определить временные окна для входов и выходов элементов. Искомое временное окно находится в результате пересечения временных окон «агрессора» и «жертвы». Для заданной неисправности - определенной пары «агрессор-жертва» метод позволяет установить необходимые значения на взаимодействующих линиях и распространить влияние неисправности до внешнего выхода. При генерации тестов используется 11-значный алфавит и модификация метода PODEM на этапе доопределения (justification). Разработанный алгоритм не гарантирует построение тестов вследствие ограничений для условий распространения

эффектов неисправностей и используемой (11-значной) логической системы.

В [6] авторы развили модель задержки пути в комбинации с критическим путем и множеством источников взаимодействующих с этим путем. Здесь в качестве основы используется метод генерации тестов для неисправностей типа «задержка распространения сигнала» без фазы доопределения (justification). А на этапе доопределения (justification) применяется генетический алгоритм, который в процессе построения теста использует временные характеристики. Далее это направление получило развитие в [7], где предложено решение задачи построения тестов в случае многих «агрессоров», воздействующих да определенный путь. Здесь строится граф импликации, который учитывает логическую и структурную информацию схемы для проверки конфликтных ситуаций в процессе генерации тестов. При поиске тестовых наборов используется модификация метода PODEM.

Работа [8] посвящена тестированию неисправностей "crosstalk" - индуцированных задержек. Алгоритм построения тестов основан на генерации критических путей на основе статического временного анализа схемы. Критерий устойчивой тестируемости используется при проверке чувствительности путей. Для чувствительного пути активируются пары «агрессор-жертва» таким образом, чтобы максимизировать воздействие агрессора на путь и вызвать задержку распространения сигнала по этому пути.

В [9] предложен генератор тестов на основе единичной точной модели задержки. Здесь активизируется подпуть для выполнения условий переходов сигналов. Известная модель неисправности типа «задержка» модифицируется для тестирования с - индуцированных задержек. Для сокращения множества анализируемых неисправностей выполняется препроцессорная обработка для критических путей.

Авторы [10] разработали метод генерации тестов на основе PODEM. Для тестирования "crosstalk" необходимо подать на внешние входы необходимые переходы и обеспечить распространение влияния неисправности на внешний выход. Каждая из этих задач формулируется отдельно, а конечное решение получается в результате пересечения решений указанных подзадач.

В [11] при генерации тестов для "crosstalk" - индуцированных задержек используется алгоритм построения тестов для неисправностей типа «задержка», основанный на троичном алфавите, где неопределенное значение применяется для моделирования переходов сигналов, которые вызываются перекрестными неисправностями.

В работе [12] при решении задачи построения тестов для "crosstalk" применяется 0-1 целочисленное программирование на основе традиционных методов генерации тестов для константных неисправностей. Максимальная активация агрессора формулируется в терминах линейного программирования, а распространение влияния неисправности решается традиционными средствами.

Следует отметить, что все вышеперечисленные работы при генерации тестов используют механизмы «возврата назад» по схеме, которые требуют значительных вычислительных ресурсов.

С другой стороны существует группа методов, которые основаны на моделировании и не требуют «возврата назад». К ним относится, например, работа [13], где моделирование применяется для тестирования перекрестных неисправностей. Здесь используется алгоритм редакции для неисправностей – кандидатов, что позволяет получать компактное множество целевых неисправностей за счет исключения тех неисправностей, которые никогда не могут быть активированы и обнаружены. При этом используется последовательный симулятор для моделирования задержек путей. Кроме этого используется аналоговая макромодель для вычисления задержек сигналов вследствие перекрестного взаимодействия. Здесь с каждой неисправностью ассоциируется одна пара агрессор-жертва. Входные наборы в процессе поиска генерируются случайным образом.

Авторы [14] предложили симулятор неисправностей, которые включают одну линию-жертву и множество линий-агрессоров. При генерации тестов используется двоичное логическое моделирование и в качестве основы используются тесты для константных неисправностей. Следует отметить, что методы, основанные на моделировании, требу ют хороших эвристик для генерации входных наборов и получаемые тестовые последовательности, как правило, имеют большую длину.

Генерация тестов для неисправностей индуцированные задержки

Генерация проверяющих тестов для этих несправностей отличается от построения тестов для несправностей 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 предыдущей работе [1].

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.

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 обычнонаходится 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 временным окном потенциальной линии- агрессора. Если эти окна пересекаются, то эта пара жертва-агрессор заносится в множество перекрестных 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].

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

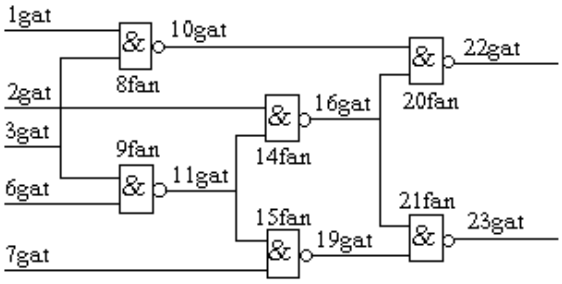

Рисунок 1 - Схема С17 каталога ISCAS85.

Всего для схемы рис.1 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.

Моделирование перекрестных неисправностей типа задержка

Моделирование перекрестных неисправностей выполняется на основе событийного логического моделирования в многозначном алфавите [15]. 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.

Генетический алгоритм генерации теста

Генетические алгоритмы широко применяются при построении проверяющих тестов для константных неисправностей и более сложных моделей, например, задержек распространения сигналов [16] в силу 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 фитнесс-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.

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

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

          }
        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 фитнесс-функции;
  }
for(каждой последовательности)
{
 if(последовательность обнаруживает неисправность из F_R)
       {
 Включение последовательности в популяцию;
 Исключение проверенной неисправности из F_R;
        } } }
```

Выводы

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

Список литературы

1. Скобцов Ю.А. Генерация тестов для неисправностей типа индуцированные импульсы / Ю.А. Скобцов, В.Ю. Скобцов, Нассер Іяд К.М. // Інформаційно-керуючі системи на залізничному транспорті. – 2010 – №4. – С.27-29.

2. 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.

3. W.Y.Chen.Analytic Models for Crosstalk Delay and Pulse Analysis under Non-Ideal Inputs/W.Y.Chen, S.K.Gupta,M.A.Breuer.-Proc.of the Int'1.Test Conf., pp. 809-818, Nov.1997.

4. W.Y.Chen.Test generation for Cross-Induced Delay in Integrated Citcuits /W.Y.Chen, S.K.Gupta,M.A.Breuer.-Proceedings of IEEE International Test Conference, pp.191-200, October 1999.

5. W.Y.Chen.Test generation for Cross-Induced Faults: Framework and computational results /W.Y.Chen, S.K.Gupta,M.A.Breuer.-Journal of Electronic Testing: Theory and Applications, vol.16, pp.17-28, Feb.2002.

6. A.Krstic. Delay Testing Considering Cross-Induced Effects/ Krstic, J.-J.Liou, Y.-M.Jiang, K.-T.Cheng.-Proceedings of International Conference,2001.

7. X.Bai.HyAC: A Hybrid Structural SAT Based ATPG for Crosstalk / X.Bai, S.Dey, A.Krstic.-Proceedings of International Conference, 2003, pp.112-121.

8. A.Arunachalam. A Novel Algorithm for Testing Crosstalk Induced Delay Faults in VLSI Cicuits/Aniket and R.Arunachalam.- Proceedings of International Conference on VLSI Design, pp.479-484, 2005.

9. H.Li. Selection of Crosstalk-induced Faults in Enhanced Delay test/ H.Li,X.Li.-Journal of Electronic Testing: Theory and Applications, Vol.21,N0.2, pp.181-195, April 2005.

10. Palit A.K. Test Pattern Generation for Crosstalk Faults in DSM chips using Mofified PODEM/ Palit A.K., Duganapali K.K., Anheier W.-TuZ, pp.41-45. 11. Sunghoon Chun. XPDF-ATPG: An Efficient Test Pattern Generation for Crosstalk-Induced Faults/ Sunghoon Chun, Yongjoon Kim, Myuang-Hoon Yang, Sungho Kang.- 17th Asian Test Symposium, pp.83-88, 2008.

12. Kunal P.Ganeshpure. On ATPG for Multiple Aggressor Crosstalk Faults in Presence of Gate Delays / Kunal P. Ganeshpure, Sandir Kundu.- Proceedings of IEEE International Test Conference, pp.1-7, October 2007.

13. Shweta Chary. Automatic path delay test generation for combined Resistive Vias Resistive bridges and Capacitive Crosstalk delay faults / Shweta Chary,Michael L. Bushnell.-Proceedings of the 19th International conference on VLSI Design, pp.413-418, 2006.

14. Marong Phadoongsidhi. SCINDY: Logic Crosstalk Delay Fault Simulation in Sequential Circuits / Marong Phadoongsidhi, Kewal K. Saluja.- 18th Internartional Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design (VLSID'05), pp.820-823, 2005.

15. Скобцов Ю.А. Логическое моделирование и тестирование цифровых устройств / Ю.А.Скобцов, В.Ю.Скобцов. – Донецк:ИПММ НАНУ, ДонНТУ, 2005. – 436с.

16. Скобцов Ю.А. Основы эволюционных вычислений / Ю.А. Скобцов. – Донецк: ДонНТУ, 2008. – 326 с.

Надійшла до редколегії 10.10.2011

Ю.О. СКОБЦОВ, В.Ю. СКОБЦОВ, ІЯД К.М. НАССЕР
Донецький національний технічний університет

Побудова перевіряльні тестів для перехресних несправностей типу затримка

Розглядається проблема побудови перевіряльник тестів для "перехресних несправностей" типу затримка, що характерні для глибокого субмікронного проектування елементної бази сучасних комп'ютерних систем. Використовується багатозначне логічне моделювання й генетичний алгоритм генерації тестів.

перехресні несправності, перевіряльні тести, генетичний алгоритм, багатозначне логічне моделювання

Y.A. SKOBTSOV, V.Y. SKOBTSOV, IAD K.M. NASSER
Donetsk National Technical University

Check Test Generation for Delay Crosstalk Faults

The checking tests construction problem for delay crosstalk faults is examined. These faults inherent for the deep submicrometer design of element base for the modern computer systems. Multiple-valued logical design and genetic checking tests generation algorithm are used.

crosstalk faults, checking tests, genetic algorithm, multivalued logic simulation